Pål Johan From

Off-Shore Robotics

Robust and Optimal Solutions for Autonomous Operation

Thesis for the degree of Philosophiae Doctor

Trondheim, May 2010

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics
and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Engineering Cybernetics

ISBN 978-82-471-2154-2 (printed ver.)
ISBN 978-82-471-2155-9 (electronic ver.)
ISSN 1503-8181
ITK report 2010-8-W

Doctoral theses at NTNU, 2010:96

Printed by NTNU-trykk

Summary

The vast majority of research in the field of robotics has over the last few decades shifted from industrial robots—in the sense of robots mounted in a structured environment such as a factory floor—to robots operating in unstructured and harsh environments. Even though industrial robotics has become a mature research field we believe that there is still room for progress and improvement. In fact, we show this through both theoretical advances and experimental results in this thesis. However, the main focus of most researchers today has shifted towards autonomous robots and robots in unstructured environments, and this is also the main focus of the work presented here.

This thesis is concerned with the borderline between the mature technology that includes conventional industrial robots and robots operating in unstructured and harsh environments, an area that is still undergoing considerable advances. The oil fields of the future will have to adopt solutions from both mature and evolving technologies—including the borderline between them—and we will use this application to illustrate the practical importance of the theoretical results throughout the thesis. Several tasks to be performed by the robot on the oil fields of the future resemble the tasks performed by industrial robots on the factory floor in thousands of factories around the world today. In this sense we may consider this a mature and robust technology. At the same time the robots will have to work in an unstructured environment with little or no direct human intervention and autonomous operation is required also for non-routine tasks. In this sense, the oil platforms of the future present us with an interesting case study.

We believe there to be two main issues that need to be addressed before partially or completely robotised oil platforms will see the light of day. The first is concerned with robustness. Robotic solutions will only be applied to tasks where the efficiency, accuracy, repeatability, and robustness surpass those of the human operator performing the same tasks. Remotely located oil platforms, and especially the ones located in sensitive areas such as the Barents Sea, are characterised by strict legislative standards to protect the environment and the wildlife. Any installation involving petrochemicals such as oil and gas will have to show for robust and reliable solutions at every stage of the operation. Thus, any operation involving robots will have to meet very high standards when it comes to robustness and fault tolerance. This imposes great challenges on the oil companies that are to operate in these areas.

The second issue is concerned with the effectiveness and cost efficiency of the operation. An experienced human operator has an incredible capability to find efficient solutions to both routine tasks and unexpected occurrences. The robotics system should also strive to solve any task in an optimal manner. Effective and cost efficient operation is vital to

be able to justify the high investment in research and installation related with robotised oil fields. Thus, for these systems to be interesting for the oil companies the daily operation should not only be more robust compared to human operators, it should also be more cost efficient to guarantee that the investment in expensive robotic equipment pays off.

When it comes to remotely located oil fields, effectiveness and robustness are very much related. The main economical risk involved with the operation of oil platforms today is unscheduled shutdowns. Unscheduled shutdowns, as well as planned maintenance shutdowns, should be made as short as possible and if possible avoided. A robust solution with less chance of failure is thus something the oil companies strive for also with the economical gain as a motivation.

This thesis is divided into four main parts. Part I is concerned with a large class of robotic systems that will play a very important role in the operation of remotely located oil platforms, namely vehicle-manipulator systems. One application of such systems is subsea installations where humans do not have direct access. The use of robots mounted on a underwater vehicles is believed to be the on-shore operator's main tool for surveillance and operation of these fields. We are mainly concerned with the mathematical modelling of a large class of systems, including vehicle-manipulator systems. The main contribution of this part is the derivation of the dynamics of a general class of vehicle-manipulator systems that also allows for joints that cannot be represented with generalised coordinates. These types of joints are often subject to singularities in the representation, but we use Lie groups and Lie algebras to represent the transformation between the local and global velocity variables and thus obtain a singularity-free formulation of the dynamics. The papers that are published in this part serve as a detailed study of vehicle-manipulator systems and are also intended to introduce these results to some relevant research areas where a singularity-free formulation is not normally adopted. We show that with our formulation we obtain a set of dynamic equations with the same complexity as the conventional Lagrangian approach but without singularities. The joints are classified depending on what Lie group we use to represent the configuration space so we can easily build a library of joints types for easy implementation in a simulation environment.

While Part I is mainly concerned with robustness in the sense that a suitable mathematical representation is chosen, Part II deals with robustness of the manipulator design. Specifically we address the problem of joint failure, i.e. when the joint loses its actuation and becomes a passive joint. This is an extremely serious situation as the passive joints in general cannot be controlled, and external forces, such as gravity or inertial forces, may cause the manipulator to collapse. This can result in severe damage to the robot's surroundings. Based on a geometric approach, we thus analyse in detail the effects of joint failure on serial and parallel manipulators. We find that for serial manipulators this should be dealt with in the control of the robot once a joint failure is identified. For parallel manipulators, however, this should be dealt with in the design of the manipulator. We present a complete set of rules on how to choose the active and passive joints in a parallel manipulator in order to guarantee fault tolerance.

Maintenance tasks on oil platforms are very important and especially for platforms situated in high sea, cold locations, and rough environments in general. The corrosion due to the salt water is for example very damaging to both the platform construction and the process area of the platform. High pressure water blasting is thus essential for maintenance and cleaning of the equipment. High pressure water blasting is also used for removing ice

and preparing the surfaces to be painted. Painting the huge surfaces of the platform with frequent intervals is another very time consuming task that needs to be performed by the robots. In Part III we show how we can improve efficiency for robots that are to perform these tasks by introducing an extended definition of functional redundancy. All the tasks above are so called pointing tasks, i.e. tasks where only the direction of the robot tool is of concern and not the orientation. We extend this definition to also allow a small error in the orientation of the robot tool. For these tasks a small error in the orientation will not affect the quality of the job, but—as we illustrate through both theoretical and experimental results—this allows us to substantially reduce the time needed to perform the task. We show how to cast the problem of finding the optimal orientation error into a convex optimisation problem that allows us to find the solution in real time. This makes the approach suitable for several tasks performed by the robotic systems on oil platforms, but also for spray paint and welding applications in factory installations.

The process area on an oil platform is very complex and the use of redundant manipulators to get access to every part of the robot is inevitable. The inverse kinematics of redundant robots, however, is challenging due to the infinite number of solutions to the problem. There are also other robots that do not have a known analytical solution to the inverse kinematics problem. One example is robots with complex geometry, which often occurs when we put the cables connecting the tool and the base on the inside of the manipulator structure. Similarly to kinematically redundant manipulators there is in this case no known solution to the inverse kinematics problem, which thus needs to be solved numerically. In Part IV we present a set of iterative solutions to the inverse kinematics problem. We divide the problem into several sub-problems that can be solved analytically. Due to the analytical solution of every sub-problem we are able to solve the inverse kinematics problem very efficiently. The approach is also very robust compared to conventional Jacobian-based methods when the initial point is far from the solution. We also present an alternative formulation of the gradient method where we solve both the problem of finding the gradient and the search along the gradient analytically. Even thought our "gradient" is only an approximation of the actual gradient, the approach is computationally very efficient and a solution is found very quickly.

This thesis addresses several different topics in robotics. All the results presented can be applied to off-shore robotics, but there are also other applications where the results are indeed applicable. As the project has progressed we have discovered several of these alternative applications and in some cases the theory presented is just as relevant in areas other than where it was originally intended. We have included several examples including space robotics and industrial spray paint robots to illustrate this, and we believe this diversity in terms of applications strengthens the theoretical results presented throughout the thesis.

Preface

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy at the Norwegian University of Science of Technology. The research presented in this thesis is the results of my doctoral studies in the period August 2006 through November 2009.

During these years I was fortunate enough to spend time at several different research institutions. The first year I was with the Department of Engineering Cybernetics, NTNU where I also started my master studies five years earlier. I also stayed seven months at Hong Kong University of Science and Technology followed by 13 months at University of California, Berkeley. All these research institutions have influenced the outcome of my research in ways that are all very evident in the different parts of this thesis.

The last months of my doctoral studies, I was at ABB Strategic R&D for Oil, Gas and Petrochemicals in Oslo. Here I had access a well equipped lab with three robots. This gave me valuable experience with the robots as well as the possibility to validate the theoretical results in the lab.

This work has been sponsored financially by Norwegian Research Council and the TAIL IO project. I am grateful for their continuous funding and support of this project. The TAIL IO project is an international cooperative research project led by Statoil and an R&D consortium consisting of ABB, IBM, Aker Solutions and SKF. The aim of the project was to deliver first class innovations to extend the lifetime of existing oil and gas fields and to look at possibilities for improved solutions for oil fields of the future. I was particularly involved with finding robotic solutions for autonomously operating the oil fields of the future in a safer and more cost efficient manner.

Trondheim, February 2010 Pål Johan From

Acknowledgement

The research presented in this thesis is the result of about three years of hard work. During these years I have been fortunate to interact with a great number of people who have shaped the outcome of this thesis in many different ways.

First of all I would like to thank my supervisor Professor Jan Tommy Gravdahl for giving me the opportunity to study the world of robotics in such depth. From the very beginning he has shown me the way and kept me out of all the pit-falls that I would otherwise dig myself into if it had not been for his firm guidance. At the same time he has allowed me to evolve my own ideas and interests. He has also encouraged me to pursue my early results to get a deeper understanding and more complete and rigid results. His experience as a researcher has been of incredible value to me, from the simple formalities of writing papers to in-depth mathematical discussions. I am also grateful for how he has laid out a path of my future.

I would also like to thank the rest of the academic staff at ITK. I have had many interesting discussions with many of you. I would especially like to thank Professor Kristin Y. Pettersen for encouraging discussions and for sharing her knowledge and experience with me. She has encouraged me to write many of the papers included in this work and her detailed feed-back has been very valuable.

There are several people at ITK who have contributed to this thesis by allowing me to focus on my academic work. I owe great thanks to Tove, Unni and Eva for answering all my e-mails so promptly and helping me out with the different administrative tasks.

I am also grateful to Professor Zexiang Li for seven wonderful months at Hong Kong University of Technology. I got to experience a different approach to the field of robotics and I also learned how the wonderful world of Lie groups can simplify so many aspects of robot modelling and control. I also got to know several of Professor Li's students and I am very grateful for the warm welcome and the way they showed me around in China and taught me about Chinese culture. I hope we will stay in touch also in the future.

I also stayed 13 months at University of California, Berkeley and will be ever thankful to Professor Shankar Sastry for giving me this opportunity. This was a very inspiring time for me, which is very much thanks to post.doc. Vincent Duindam who took good care of me. In addition to helping me out with the administrative work, showing me the campus and the city, and taking me to lunch, I learned a lot from him and in particular how nice it is to live in a singularity free world. I think the few months we worked together resulted in some very nice results and papers. I would also like to thank Assistant Professor Pieter Abbeel for introducing me to the world of learning in robotics and patiently guiding me in my work in this field. This was a completely new field for me which I have learned to admire thanks to Pieter's classes and his enthusiastic supervision.

I have performed several experiments that would not have been possible without the help of several different people. I am very grateful to “Erik Bye” and its crew and Redningsselskapet AS for letting me sail with them. I obtained most of the measurements that I have used in my work during our trip. Thanks also to Professor Thor I. Fossen for sharing his knowledge in ship modelling and for introducing me to several people, it turned out very valuable for me. Many thanks also to ABB and especially Charlotte Skourup, Johan Gunnar, David Anisi and Tommy Lillehagen at Strategic R&D in Oslo, Norway. It was great staying with this group the last months of my studies and I am especially thankful that, with their expertise, I got all my experiments done in such a short time.

This project would never be possible without the financial support of several generous contributors. First of all the Nowegian Research Council contributed generously as well as all the participants in the TAIL IO project, including Statoil and ABB. I would also like to thank the Strategic University Program on Control, Information and Communication Systems for Environmental and Safety Critical Systems, and the Strategic University Program on Computational Methods in Nonlinear Motion Control for generous contributions that made it possible to spend so much time abroad. Thanks also to “Forsknings og Undervisningsfondet i Trondheim” and Office of International Relations at NTNU for financial contributions.

I would also like to thank my mother Reidun, my father Johan, and my sister Tale. Every aspect of my Ph.D has been discussed in detail over family dinners and I know that without the support of my family I would not have started this journey. I particularly remember the moment when I decided that this was what I wanted to do. We were at our cabin in the mountains in the Easter of 2006 and after hours of skiing, conversation and plenty of advice from my parents I decided that this was how I wanted to spend the next three years of my life. I am very happy we took that ski trip.

Finally I would like to thank Carla who has been by my side every day these three years. She has travelled with me first to Trondheim, then to Hong Kong and Berkeley, and then back to Oslo again. Always positive. Always smiling. I hope and believe that these three years have been as fruitful to her as they have to me. I am very grateful that I have met such a positive and energetic person who stands by my side every day of the year, any place in the world.

Contents

Publications

The material presented in this thesis is based on the publications listed below. The chapters in which the papers occur are included for each paper. If the papers in large or completely are covered by other publications they are omitted from the thesis. The papers that are included and thus are to be considered a part of this Ph.D dissertation are market with an asterisk.

Part I - Singularity-Free Vehicle-Manipulator Modelling

- **Singularity-Free Dynamic Equations of Vehicle-Manipulator Systems*** (From et al., 2010a),
  P. J. From, V. Duindam, K. Y. Pettersen, J. T. Gravdahl and S. Sastry,
  Simulation Modelling Practice and Theory, Vol. 18, No. 6, 2010.
  Chapter 2.

- **Singularity-Free Dynamic Equations of AUV-Manipulator Systems** (From et al., 2010d),
  P. J. From, K. Y. Pettersen and J. T. Gravdahl,
  Symposium on Intelligent Autonomous Vehicles, 2010.
  This paper is fully or partially covered by From et al. (2010a), see Chapter 2.

- **Modeling and Motion Planning for Mechanisms on a Non-Inertial Base*** (From et al., 2009a),
  P. J. From, V. Duindam, J. T. Gravdahl and S. Sastry,
  IEEE International Conference on Robotics and Automation, 2009.
  Chapter 3.

- **Motion Planning and Control of Robotic Manipulators on Seaborne Platforms*** (From et al., 2009b),
  P. J. From, J. T. Gravdahl, T. Lillehagen and P. Abbeel
  Submitted to Control Engineering Practice.
  Chapter 4

- **On the Influence of Ship Motion Prediction Accuracy on Motion Planning and Control of Robotic Manipulators on Seaborne Platforms** (From et al., 2010b),
  P. J. From, J. T. Gravdahl and P. Abbeel
  IEEE International Conference on Robotics and Automation, 2010.
  This paper is fully or partially covered by From et al. (2009b), see Chapter 4.

- **On the Boundedness Property of the Inertia Matrix and Skew-Symmetric Property of the Coriolis Matrix for Vehicle-Manipulator Systems*** (From et al., 2010f),
  P. J. From, I. Schjølberg, J. T. Gravdahl, K. Y. Pettersen and T. I. Fossen
  Submitted to Systems and Control Letters.
  Chapter 5.

- **On the Boundedness and Skew-Symmetric Properties of the Inertia and Coriolis Matrices for Vehicle-Manipulator Systems** (From et al., 2010g),
  P. J. From, I. Schjølberg, J. T. Gravdahl, K. Y. Pettersen and T. I. Fossen
  Symposium on Intelligent Autonomous Vehicles, 2010.
  This paper is fully or partially covered by From et al. (2010f), see Chapter 5.

- **Singularity-Free Dynamic Equations of Spacecraft-Manipulator Systems** (From et al., 2010c),
  P. J. From, K. Y. Pettersen and J. T. Gravdahl,
  IFAC Symposium on Automatic Control in Aerospace, 2010.
  This paper is fully or partially covered by From et al. (2010a), see Chapter 2.

- **Singularity-Free Formulation of the Dynamically Equivalent Manipulator Mapping for Free-Flying and Free-Floating Space Manipulators*** (From et al., 2010e),
  P. J. From, K. Y. Pettersen and J. T. Gravdahl,
  To be submitted to AAIA Journal of Guidance, Control, and Dynamics
  Chapter 6.

- **Singularity-Free Formulation of the Dynamically Equivalent Manipulator Mapping for Space Manipulators** (From et al., 2011),
  P. J. From, K. Y. Pettersen and J. T. Gravdahl,
  To be submitted.
  This paper is fully or partially covered by From et al. (2010e), see Chapter 6.

Part II - Robust Manipulator Design

- **A Geometric Approach to Handling Torque Failure in Serial and Closed Chain Manipulators with Passive Joints*** (From and Gravdahl, 2009b),
  P. J. From and J. T. Gravdahl,
  Submitted to ASME Journal of Mechanical Design
  Chapter 7.

- **On the Mobility and Fault Tolerance of Closed Chain Manipulators with Passive Joints** (From and Gravdahl, 2008e),
  P. J. From and J. T. Gravdahl,
  Modeling, Identification and Control, Vol. 29, No. 4, 2008.
  This paper is fully or partially covered by From and Gravdahl (2009b), see Chapter 7.

- **Fault Tolerance of Parallel Manipulators with Passive Joints** (From and Gravdahl, 2009a),
  P. J. From and J. T. Gravdahl,
  IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, 2009.
  This paper is fully or partially covered by From and Gravdahl (2009b), see Chapter 7.

Part III - Functional Redundancy

- **Representing Attitudes as Sets of Frames*** (From and Gravdahl, 2007b),
  P. J. From and J. T. Gravdahl,
  American Control Conference, 2007.
  Chapter 8.

- **On the Equivalence of Orientation Error and Positive Definiteness of Matrices*** (From and Gravdahl, 2008d),
  P. J. From and J. T. Gravdahl,
  International Conference on Control, Automation, Robotics and Vision, 2008.
  Chapter 9.

- **A Real-Time Algorithm to Determine the Optimal Paint Gun Orientation in Spray Paint Applications*** (From and Gravdahl, 2010b),
  P. J. From and J. T. Gravdahl,
  IEEE Transactions on Automation Science and Engineering.
  Chapter 10.

- **Representing Sets of Orientations as Convex Cones** (From and Gravdahl, 2009c),
  P. J. From and J. T. Gravdahl,
  IEEE International Conference on Robotics and Automation, 2009.
  This paper is fully or partially covered by From and Gravdahl (2010b), see Chapter 10.

- **Real-time Optimal Trajectory Planning for Robotic Manipulators with Functional Redundancy** (From and Gravdahl, 2008f),
  P. J. From and J. T. Gravdahl,
  International Conference on Industrial Technology, 2008.
  This paper is fully or partially covered by From and Gravdahl (2010b), see Chapter 10.

- **Optimal Paint Gun Orientation in Spray Paint Applications - Experimental Results*** (From and Gravdahl, 2010a),
  P. J. From, J. Gunnar, and J. T. Gravdahl,
  Submitted to IEEE Transactions on Automation Science and Engineering.
  Chapter 11.

Part IV - Inverse Kinematics of Robotic Manipulators with no Closed Form Solution

- **Iterative Solutions to the Inverse Geometric Problem for Manipulators with no Closed Form Solution*** (From and Gravdahl, 2008c),
  P. J. From and J. T. Gravdahl,
  Modeling, Identification and Control, Vol. 29, No. 3, 2008.
  Chapter 12.

- **General Solutions to Functional and Kinematic Redundancy** (From and Gravdahl, 2008b),
  P. J. From and J. T. Gravdahl,
  Modeling, Identification and Control, Vol. 29, No. 2, 2008.
  This paper is fully or partially covered by From and Gravdahl (2008c), see Chapter 12.

- **General Solutions to Functional and Kinematic Redundancy** (From and Gravdahl, 2007a),
  P. J. From and J. T. Gravdahl,
  IEEE Conference on Decision and Control, 2007.
  This paper is fully or partially covered by From and Gravdahl (2008c), see Chapter 12.

- **Considering Passive Joints as Functional Redundancy** (From and Gravdahl, 2008a),
  P. J. From and J. T. Gravdahl,
  IFAC world congress, 2008.
  This paper is fully or partially covered by From and Gravdahl (2008c), see Chapter 12.

Other Publications

- **Trends in Research and Publication: Science 2.0 and Open Access** (Breivik et al., 2009),
  M. Breivik, G. Hovland and P. J. From,
  Modeling, Identification and Control, Vol. 30, No. 3, 2009.
  Not included.

Note to the Reader

Except a few cases noted below all the papers included in this thesis are exact reproductions of the published or submitted papers referred to in the publication list. As a result of this the notation is not necessarily consistent in the different parts of the thesis. For each paper we have chosen the notation that we believe is the most commonly used and easiest to read for the target audience.

Several different authors have participated in the writing and proof-reading of these papers. The language in this thesis and some papers is UK English, but in some papers, especially in Part I, we use US English.

All papers in this thesis are exact reproductions of the published and submitted papers in the publication list, except from the following:

- Obvious typing mistakes are corrected.
- Some black-and-white figures have been substituted with colour figures.
- Equation (68) (in the original paper) has been removed from From and Gravdahl (2007b).
- In Chapter 10 Equation (10.113) has been corrected so to correspond with (10.22).
- In some of the papers in Part III we have changed the way we write the quaternion product so that all papers use the same convention.
- Chapter 5 has been updated to a more recent version than the one originally submitted.

Certain sections are more or less identical in two or more of the papers that are included in this thesis. This is typically introductions and background material that are similar in the different papers. We list some parts that are covered more than once so that the reader can skip or read quickly through these parts. These parts are included for the sake of completeness of the papers.

- Part I: For the papers in Part I the Introduction (Section 2.2) will to some extent cover introductions of Chapters 3, 4 and 6. The same is the case for Section 2.3 which covers much of the mathematical background of vehicle-manipulator modelling in the same chapters.
- Part III: In this part the background material on "representing rotations" and "quaternion volumes" is very much the same in all the papers. The papers in this part are included chronologically as one paper is in many ways a natural extension of the previous paper. Chapter 10 serves as a summary of the theoretical results in this part.

Chapter 1

Introduction

1.1 Robotic Solutions in Off-Shore Applications

There is a strong consensus that off-shore oil fields are undergoing a fundamental change in the way they are operated: from mainly human operation to a completely or partially automated operation where robots perform the different tasks and with humans mainly as observers and supervisors (Skourup and Pretlove, 2009). This change has not yet taken effect, but the increasing interest, both through research projects and financial investment from oil companies and robot manufacturers, suggests that this change will come in the very near future.

There are two main motivations behind this fundamental change. The first is increased productivity and cost efficient solutions due to more reliable solutions, less manpower, and smaller platforms through the removal of living facilities. The second motivation—and maybe the most important one—is that the oil fields of the future cannot be built with today's technology. Challenges such as deeper water, remote locations, and challenging weather conditions require new technology.

The transition from platforms operated by humans to more autonomous solutions will involve many interesting aspects, especially when it comes to utilising solutions taken from industrial robots in unstructured and far more challenging areas. In this thesis we discuss several topics in robotics that need to be solved before this transition can take place.

1.1.1 Oil Fields of the Future

On existing oil fields robotic solutions are already used for several tasks, but this is in general limited to very specialised niche applications or tasks that cannot or are too dangerous to be performed by humans. One example is the use of remotely operated underwater vehicles for inspection of subsea processes. This technology is now developed to a level where it can be considered functional as well as relatively robust. For topside[1] applications, however, we have not seen the same interest in developing robotic solutions. This is

[1] Topside means offshore oil and gas installations and ships above the water level.

Figure 1.1: The offshore Statoil Troll A gas platform located in the North Sea. Note the large living facilities on the right, which is required due to the large number of workers on the platform. Courtesy ABB Strategic R&D for Oil, Gas and Petrochemicals.

mainly because, up to today, this has not been cost efficient, nor has it improved the safety aspects of the operation.

This has, however, changed over the last decade and new technology is believed to introduce robotic solutions to a wider range of applications. There are many reasons for this. Firstly, new and cheaper technology has made the use of robotic solutions more cost efficient than flying human operators to distant fields. The relatively high one-time investment will pay off due to reduced salaries to human workers. Secondly, robotic solutions are more reliable and less error prone than humans and we can expect fewer and shorter shutdowns. Robots work 24 hours a day seven days a week and do so very reliably. Human workers are subject to stress and difficult working conditions which lead to mistakes and accidents. Robots thus reduce the probability of un-planned production stops which is extremely costly for the oil companies. Thirdly, most of the accessible fields close to shore have already been explored and future fields are situated in more remote and harsh environments where accessibility is low. These fields are technically challenging and require the development of new technology. Much of this relies on the use of robotic solutions. This is not only due to the extreme costs of safely transporting human workers to these areas but also because of the extreme working conditions. Many of the oil fields that are to be developed are not suited for human workers due to temperature and weather conditions, as well as the distance to on-shore facilities. For example, the Shtokman field located in the Barents Sea some 600 kilometres north of the Kola Peninsula experiences such ex-

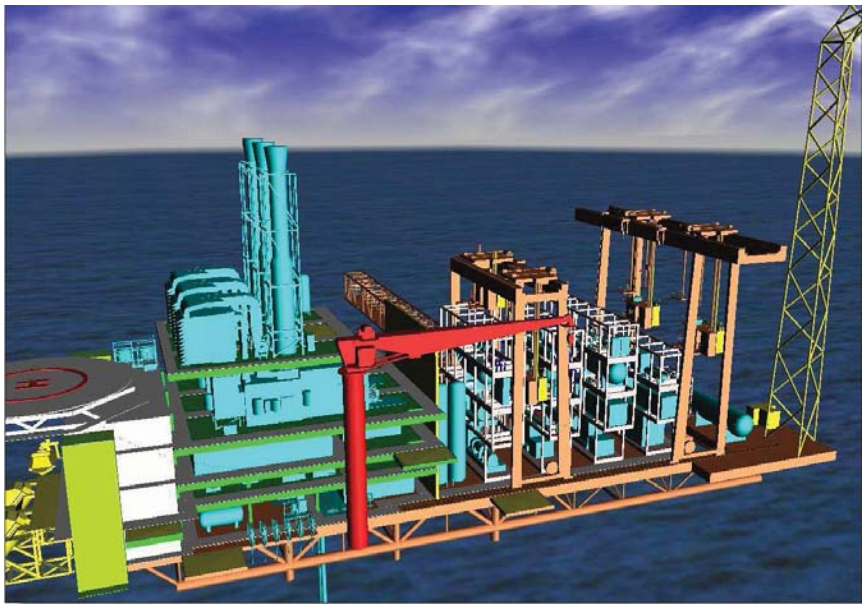


Figure 1.2: The oil platform of the future. The gantry cranes allow the robots to reach every part of the platform to perform the pre-defined or operator controlled operation. The large living facilities are removed or drastically reduced. Courtesy of TAIL IO.

treme weather conditions that for six months of the year one can not expect to frequently fly workers to and from the platforms by helicopter. Finally, the oil companies address health, safety, and the environment (HSE) as important issues. The off-shore installations are some of the most dangerous places to work due to the extreme weather, unstructured environment and high concentration of dangerous and in some cases deadly gases, such as hydrogen sulphide H_2S. There is an apparent advantage in the use of robots in these environments in order to reduce the exposure of these hazards to human workers.

History has taught us that for applications where new technology is required in order to be able to perform a given task, great effort has been put down to develop this technology and take it into use. On the other hand, when the task can be performed by a human operator and the technology is not absolutely necessary to be able to perform the task, this has not always been the case. The simple fact remains though, that for many of the tasks performed by humans today the technology is available and a robot manipulator can do the same job as the human operator. As an example, the necessary technological advances have to a large extent been developed and used for subsea and other dangerous tasks. If one can automate the process subsea, this should be achievable also topside, which is more accessible and more structured than the subsea environment.

So the question that arises is why this has not been applied to topside applications earlier. One important aspect is that a higher degree of automation naturally implies changes in technology, peoples' work patterns and workplace, and the organisational structure in general. This reorganisation is complex, much because of the robot-human interaction. Continuous monitoring and operation by the operator is required and this makes the op-

Figure 1.3: The daily operation of off-shore platforms include several challenging and potentially dangerous tasks for the human workers.

erator an integrated part of the control loop. The reorganisation also requires a complete set of new job descriptions and new competence. This requires detailed planning and preparation in order to succeed. The most important aspect, however, is probably that the transition from automating only parts of the process to completely automate an extremely complex process such as an oil field is a giant step in terms of technological advances. To develop robotic systems that can solve all the necessary tasks on an oil platform is probably possible in the very near future, but the main challenges lie in developing a system that allows for these systems to interact and work together as a whole. Completely automating the process requires both old and new technologies such as teleoperation, extreme computational power, complex control algorithms, system integration, and both the robustness and efficiency of the robotic solutions need to be improved. It is only over the last few years with advances in areas such as computational power, autonomous robots and robust solutions for robots operating in harsh environments that this technology is starting to become available.

Benefits of Autonomous Operation

There are considerable gains related to automating the oil fields of the future, both in HSE and in economic benefits. An estimate of the main advantages of automating the process is presented in Vatland and Svenes (2008) and can be summarised as

- Improved HSE Performance when automating oil rigs:
  - Reduced need for transport/shuttling
  - Reduced manning -60%

Figure 1.4: ABB-Shell collaboration room - an example of an integrated operations centre. The operator is located on-shore and has complete control of the off-shore facilities through visual tools. Courtesy ABB Strategic R&D for Oil, Gas and Petrochemicals.

- Economic benefits when automating oil rigs:[2]
  - Reduced weight of floater $\sim 36\%$
  - CAPEX[3] reduction $\sim 30\%$
  - OPEX[4] reduction $\sim 32\%$
  - Reduced emissions & chemical consumption
  - Total savings of introducing automated solutions compared to today's solution: 650 Mill USD[5]

1.1.2 Integrated Operations

Integrated Operations (IO), also known as eField, iField, Smart-Field, etc., is a fast growing research area that has been made possible mainly due to the enormous increase in the ability to collect, monitor and process real-time data. The combination of real-time and historical data from the plant and improved computational power and software represents an extremely valuable analysis tool. The main goal is to allow for collaboration across different disciplines, i.e. to allow for common access to data for different parts of the overall system, like the plant, administration, operational centres, and the control, monitoring, and collaboration rooms. Statoil defines Integrated Operations as (Vatland and Svenes, 2008):

[2] Figures from previous studies (Vatland and Svenes, 2008)
[3] Loosely speaking, CAPEX is investment in the business that adds shareholder value.
[4] Loosely speaking, OPEX is what you have to spend in order to keep your business running.
[5] Interest rate 7%, platform with 20 year field life

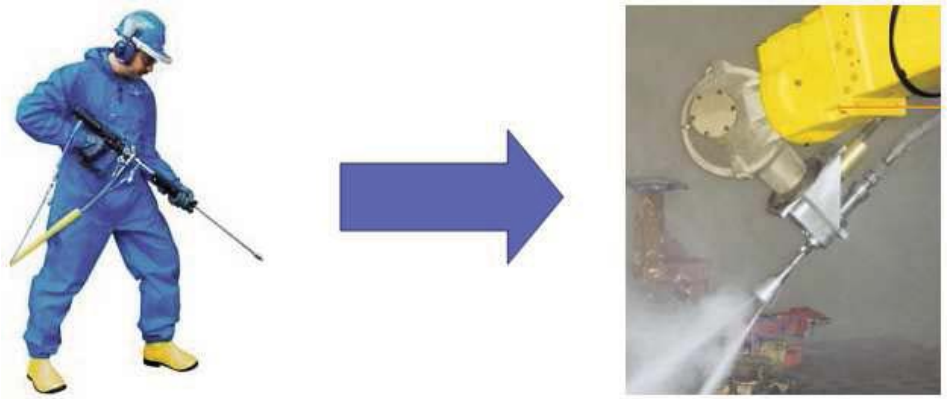


Figure 1.5: High-pressure water blasting is one relevant application that the robot can perform that will spare the human worker from potentially dangerous tasks. Courtesy ABB Strategic R&D for Oil, Gas and Petrochemicals.

"*Collaboration across disciplines, companies, organisational and geographical boundaries, made possible by real-time data and new work processes, in order to reach safer and better decisions — faster.* "

Robotics and Integrated Operations

As a part of IO, robot technology is believed to play an important role (Vatland and Svenes, 2008):

"[...] *robotics technology to supplement and extend human inspection and intervention capabilities at subsea, topside and onshore facilities. The objective is to develop solutions that combine tele-robotics and advanced visualisation to enable remotely operated inspection and maintenance operations, as well as to identify and close technology gaps*."

The following advantages are recognised in utilising robotics together with IO in future plants

- Future Plants
  - Modular, interchangeable equipment arrangement with vertical access routes.
  - Compact process technology.
  - Direct access to equipment status for remote inspection by use of Remote Assisted Tools (RATs).
  - Mobile decks give enhanced access to equipment.
- Benefits
  - HSE performance improvement.
    - Remote operation - Hazardous work done by robots.
  - OPEX reduction.

    - Considerable reduction in offshore man hours.
    - Campaign based maintenance - shortened time window.
  - CAPEX reduction.
    - Compact, modular process equipment on one deck.
    - Shortened fabrication- & project execution-period.
  - Increased flexibility.
    - Reconfigurations made quickly and at low cost.
    - Process modules can be “shared” between installations.

1.1.3 The TAIL IO Project

The TAIL IO project is an international cooperative research project led by Statoil and an R&D consortium consisting of ABB, IBM, Aker Solutions and SKF. The aim of the project is to deliver first class innovations to support Statoil in achieving their overall goals for extending the lifetime of Statoil’s oil and gas fields:

- increase daily production by at least 5 percent by reducing production losses caused by operational failure, maintenance stops and inadequate equipment performance
- reduce operating, construction and maintenance costs by 30 percent
- reduce the number of unwanted incidents relating to health, safety and the environment (HSE)

One out of six subproject is concerned with robotics technology with the intention to supplement and extend human inspection and intervention capabilities at subsea, topside and on-shore facilities. The objective is to develop solutions that combine telerobotics and advanced visualisation to enable remotely operated inspection and maintenance, as well as to *identify and close technology gaps*. The Tail IO project was supported by the Norwegian Research Council.

1.2 Robust Solutions

Robustness is the property describing the ability to operate continuously without failure under a wide range of conditions (Soanes and Stevenson, 2008). A mechanism, system or design is said to be robust if it is capable of coping well with variations in its operating environment with minimal or no damage, alteration or loss of functionality. The design of robust robotic systems is one of the most challenging topics in robotics, and in unstructured environments with unpredictable and random variations this problem becomes even harder.

It is impossible to design a system that is robust to all possible variations in the environment so the main issue here is to design a robotic system that is robust enough so that utilising it will lead to improvements over other existing systems. If operation is safer and more reliable using human workers no one will spend a considerable amount of money robotising the operation. Thus, robot manufacturers and developers need to search for areas where robots will improve both reliability and safety, and to do this in a cost efficient

Figure 1.6: The ABB robot test laboratory in Oslo, Norway. Courtesy ABB Strategic R&D for Oil, Gas and Petrochemicals.

manner. Off-shore oil facilities is definitely one of the areas that stands next in line to be robotised, but there is no doubt that the main concern for the oil companies, environmentalists, and governments is the robustness and reliability of this solution. Parts of Part I and the entire Part II of this thesis is thus devoted to robustness of robotised solutions on off-shore facilities.

Part I addresses the effects of mounting a standard robotic manipulator on a moving base, such as a ship, autonomous underwater vehicle (AUV), or platform. If the motion of this base is large, like on a ship in high sea, this will add non-inertial terms to the dynamic equations. If these non-inertial forces are dominant, conventional motion planning and control approaches may lead to instability or poor robustness and performance. This thus needs to be included in the control of the robot to guarantee robustness also in these cases.

If remotely located platforms are subject to weather conditions too extreme for humans, these weather conditions will also cause great challenges to ships sailing to these facilities. A completely automated off-shore field thus also needs to include solutions for ships. Robust operation of these ships is extremely critical as they may be loaded with petrochemicals such as crude oil and gas. At times, ships sail at high sea and, as we show in Part I, the inertial forces affect both the path planning and control of the robots operating on these ships. Conventional methods are thus not sufficient for robots mounted on a non-inertial base when it comes to robustness. This is discussed in detail in Part I.

Robust mathematical models of vehicle-manipulator systems is also of vital importance when it comes to implementing robust motion planning and control algorithms. In the setting of this thesis we will say that a model is robust if it is globally valid, i.e. singularity free. When a robotic manipulator is mounted on a vehicle that can rotate freely, singularities arise in the mathematical model if the Euler angles are used in the repre-

Figure 1.7: An ABB robot at work performing routine inspection of process equipment. Courtesy ABB Strategic R&D for Oil, Gas and Petrochemicals.

sentation. This is a problem when the dynamics are derived in the conventional way, for example for AUV-manipulator systems. AUVs are believed to be an integrated part of the oil fields of the future, and robust and computationally efficient models of the AUV-manipulator dynamics is thus also important to assure robustness of the oil fields of the future. This is also discussed in Part I.

Part III addresses some issues concerning joint failure. Specifically we address how to deal with torque failure, i.e. when the actuation of an active joint is lost and this joint becomes passive, also known as a free-swinging joint fault. A fault tolerant system should be able continue operation when this occurs. There are basically two ways to deal with this problem. The first is to develop intrinsically fault tolerant actuators through high redundancy, i.e. to use several smaller sub-actuators to build the actuators (Muenchhof et al., 2009; Steffen et al., 2009). The idea here is that a fault in one of the sub-actuators will not affect the overall performance of the main actuator. This technology is not yet ready and strictly speaking these actuator are not fault tolerant in the true sense. If for example power is lost to the actuator, this will affect all sub-actuators and a fault occurs.

The other way to deal with this is through redundant actuation. In this case the robotic manipulator can continue operation after a joint fault has occurred by utilising the redundant actuators. This solution, however, requires that all joints are represented by at least one redundant joint and can only be applied to parallel mechanisms. Redundant joints are expensive and reduce performance because of added weight. It is thus important to find the minimal set of joints, for which the manipulator remains operational when joint failure occurs for an arbitrary joint. This is discussed in Part III for parallel manipulators subject to torque failure. We also discuss the effect of joint failure in serial manipulators.

1.3 Optimal Solutions

Throughout the history of industry, there has been one dominating factor that has been driving the progress and technological advances more than any other. That factor is productivity. Over the last decade, one of the most important tools to improve productivity, and the one that has made the most difference in the setting of our work, is computer-based optimisation methods. Optimisation is basically the art of how to achieve more with less. This has been done as long as industry and technology has existed, but has reached a new era with the development powerful computers and software.

On the oil fields of the future, robotic solutions are to substitute many operations previously performed by humans. Removing the human worker from the platform area presents us with one great challenge: An experienced human operator has an incredible capability to notice when something is wrong. Through sight, hearing, smell and physically feeling the process area he or she can sense deviations from what is normal and trigger the alarm. To substitute this "sensing operator" with robotic solutions is a huge challenge. This area is closely related to machine learning and artificial intelligence.

The human operator is also very capable of finding efficient solutions to both routine tasks and unexpected occurrences. This is a characteristic of the human worker that we should emulate in our robotic system. For robotic or computerised systems the intelligence of humans is often substituted through optimisation methods. These range from complex algorithms and theory found in artificial intelligence to optimal solutions of simpler task performed by the robot. In this thesis we discuss several aspects concerning optimality in the setting of off-shore robotics.

For example, for a robot mounted on a ship, we can use information about the ship motion to find an trajectory planner and for optimal control. This will allow for reduced strain and tension on the robot, more accurate manipulation and reduced energy consumption. This is in fact an example where the performance of the robot should surpass that of the human operator. Anyone who has tried to write or pour coffee while sitting in a moving vehicle knows that humans are not very capable of reducing the effects of the non-inertial forces by compensating for these in any way. However, through sophisticated sensors and control algorithms we can drastically reduce the effects of the non-inertial forces by modifying the robot controller. This is discussed in detail in Part I.

In Part III we discuss the optimal solution to a large class of tasks that the robotic manipulator is to perform on off-shore platforms. Most tasks involved with the surveillance and maintenance of the platform require that some tool is attached at the end of the manipulator chain. For several of these tasks the orientation of this tool is critical, but there are

also some tasks for which it is not. This is the case with tasks such as high-pressure water blasting, spray painting, welding and holding a camera for surveillance. As these tasks constitute a large part of the tasks that are to be performed by the robots we study these in detail. We show that the time needed to perform these task can be drastically reduced if an optimal solution is applied and the time saved can thus be allocated to other tasks.

Finally, a human operator can extend his arm to reach areas of the process that are very difficult to access. This is an area where human performance greatly surpasses robots, as far as today's technology is concerned. In general it is very difficult for robots to operate in complex process areas. To get access to every part of the process, redundant manipulators need to be used and this greatly complicates the path planning and the inverse kinematics. In Part IV we discuss one solution to the inverse kinematics problem that can be applied to redundant actuators. The problem here is that kinematically redundant robots do not have an analytic solution to the inverse kinematics problem. Part IV is thus devoted to generic methods for solving the inverse kinematics problem for these kinds of robot arms. The approach is also applicable to other manipulators with complicated geometry that do not have a known solution to the inverse kinematics. One group of robots that do not have an analytical solution to the inverse kinematics problem consists of robots that are hollow on the inside. Making the robots hollow complicates the kinematics, but is a great advantage because placing the cables on the inside makes them more protected and they cannot get stuck in the platform's process area.

In the following we discuss the contributions of the four different parts in some more detail.

1.4 Part I - Singularity-Free Vehicle-Manipulator Modelling

Robust solutions are very important for vehicle-manipulator systems such as a manipulator mounted on an autonomous underwater vehicle (AUV) operating subsea, a manipulator mounted on a satellite operating in space, and for manipulators mounted on a moving vehicle in general. The main strength of such systems is that they can operate in distant and harsh environments and far away from humans who may just observe the operations or remotely control them. Such systems thus need to be reliable as direct access may be impossible, very time consuming, or extremely costly. The vehicle-manipulator system may also be part of a larger system, such as a subsea oil installation or a space station. These systems will then depend on all the sub-systems—including the robotic systems—for continuous operation and assistance both in routine operations and in emergency situations.

A robotic manipulator mounted on a moving vehicle is a versatile solution well suited for these applications and will play an important role in the operation and surveillance of remotely located plants in the very near future. Recreating realistic models of for example space or deep-sea conditions is thus important. Both for simulation and for model-based control the explicit dynamic equations of vehicle-manipulator systems need to be implemented in a robust and computationally efficient way to guarantee safe testing and operation of these systems.

We will discuss several different vehicle-manipulator systems in Part I. One example of such a system is spacecraft-manipulator systems (Egeland and Sagli, 1993; Dubowsky

and Papadopoulos, 1993; Hughes, 2002; Moosavian and Papadopoulos, 2004, 2007) which are emerging as an alternative to human operation in space. Operations include assembling, repair, refuelling, maintenance, and operations of satellites and space stations. Due to the enormous risks and costs involved with launching humans into space, robotic solutions evolve as the most cost-efficient and reliable solution. However, space manipulation involves quite a few challenges. When it comes to space manipulators robustness is, for obvious reasons, extremely important. The framework presented in Part I is well suited for modelling spacecraft-manipulator systems, which is quite different from standard robot modelling. Firstly, the manipulator is mounted on a free-floating (unactuated) or free-flying (actuated) spacecraft. Secondly, the motion of the manipulator affects the motion of the base, which results in a set of dynamic equations different from the fixed-base case due to the dynamic coupling. Finally, the free fall environment complicates the control and enhances the non-linearities in the Coriolis matrix. Especially when applying the so-called dynamically equivalent manipulator approach (Liang et al., 1998; Parlaktuna and Ozkan, 2004) the advantages of the framework proposed in Part I becomes apparent.

A second example which is of great importance in off-shore installations is autonomous underwater vehicles (AUVs) with robotic arms, or underwater robotic vehicles (URVs) (Love et al., 2004; Kitarovic et al., 2005; Antonelli, 2006; McMillan et al., 1995). This is an efficient way to perform challenging tasks over a large subsea area. Operation at deeper water and more remote areas where humans cannot or do not want to operate, requires more advanced and robust underwater systems and thus the need for continuously operating robots for surveillance, maintenance, and operation emerges.

The use of robotic manipulators on ships is another important application (Kitarovic et al., 2005; Oh et al., 2005). In the Ampelmann project (Salzmann, 2007) a Stewart platform is mounted on a ship and is used to compensate for the motion of the ship by keeping the platform still with respect to the world frame. This requires both good predictions of the motion of the ship and accurate models including the dynamic coupling between the ship and the manipulator. This can be modelled as a 2-joint mechanism where one joint represents the uncontrollable ship motion and one joint the Stewart platform. There are also other relevant research areas where a robotic manipulator is mounted on a floating base. Lebans et al. (1997) give a cursory description of a tele-robotic shipboard handling system, and Kosuge et al. (1992) and Kajita and Kosuge (1997) address the control of robots floating on the water utilizing vehicle restoring forces. Another interesting research area is macro/micro manipulators (Yoshikawa et al., 1996; Bowling and Khatib, 1997) where the two manipulators in general have different dynamic properties.

1.4.1 Euclidean and Non-Euclidean Joints

The framework that we will present i Part I is especially suited for modelling the aforementioned systems due to the non-Euclidean configuration space of the vehicle. The difference between Euclidean and non-Euclidean joints or transformations is very important in this setting so we start with a formal definition. We first need to define the terms generalised coordinates and generalised velocities.

Definition 1.1. A set of coordinates which uniquely describes the configuration of a body, or system of bodies, is called the generalised coordinates of the system.

This set is not unique, but there is normally a set that will allow an easier formulation and a deeper physical insight. For robotic manipulators with 1-DoF joints the joint positions are normally chosen. The minimum number of independent generalised coordinates needed to describe the configuration of a system is known as the *degree of freedom* or *mobility* of the system.

Definition 1.2. A generalised velocity $\dot{x}_i$ associated with the generalised coordinate x_i is defined as $\dot{x}_i = \frac{\mathrm{d}x_i}{\mathrm{d}t}$.

We note that it is not always possible to find a set of generalised velocities to describe the velocity state of the system in this way. In accordance with much of the literature we will say that a system can be written in terms of generalised coordinates if we can find a set of generalised coordinates *and* generalised velocities written as in Definition 1.2 that uniquely describe the state of the system.

We now turn to the definition of Euclidean and non-Eucliden joints.

Definition 1.3. A Euclidean joint is represented by a transformation where the state of the joint can be written in terms of generalised velocities, i.e. the position variables are written as $x \in \mathbb{R}^n$ and the velocity variables as $v = \dot{x} \in \mathbb{R}^n$ where $\dot{x} = \frac{\mathrm{d}x_i}{\mathrm{d}t}$.

All 1-DoF joints are Euclidean and thus also the most commonly found robotic joints. Also joints with only translational motion are Euclidean.

Definition 1.4. A non-Euclidean joint is represented by a transformation where the state of the joint cannot be written in terms of generalised velocities, i.e. the position variables are written as $x \in \mathbb{R}^n$ and the velocity variables as $v = S(x)\dot{x} \in \mathbb{R}^n$ for some transformation matrix $S(x)$ and $\dot{x} \neq \frac{\mathrm{d}x_i}{\mathrm{d}t}$. The velocity variables are thus not simply the time derivative of the position variables.

A spherical joint, or the attitude of a rigid body, is thus a non-Euclidean joint. We see this if we represent the position variables as the Euler angles for which the state variables cannot be written in vector form where the velocity is simply the time derivative of the position variable.

1.4.2 A Short Overview of Modelling of Mechanical Systems

There is a wide variety of approaches that serve as a starting point to derive the dynamics of mechanical systems. Common for all these is that we want to extract the components of the forces and state variables that are of interest for simulation and control, and for which the dynamics are valid. We will in brief discuss three main groups of systems for which we need to project the dynamics into the useful/valid components: 1) the principle of virtual work where the equations of motion are projected into the directions associated with the generalised velocities; 2) systems where the nonholonomic forces need to be eliminated from the equations; and 3) systems where the configuration space needs to be projected from the Euclidean space to a configuration space with a topology different than that of $\mathbb{R}^n$.

The first groups is related with d'Alembert's principle where the equations of motion are projected into the directions associated with the generalised velocities and thus the

forces and torques of the constraints are eliminated from the equations. There are many different forces present in a mechanical system, but from a simulation and control point of view we are mainly interested in how the actuator and external forces affect the accelerations of the rigid bodies. Other forces, such as forces of constraint or internal forces are not of interest when it comes to modelling for control. These forces can then be eliminated from the dynamic equations in different ways, for example using the principle of virtual work which has its roots back to d'Alembert's principle. The principle of virtual work allows us to eliminate the constraint forces without actually deriving them explicitly, which can be quite hard.

Another point of view regarding the virtual work is the interpretation that the dynamic equations are projected into directions associated with the generalised coordinates. The Newton-Euler equations can be derived from d'Alembert's principle and have shown to be very useful in robotics, and especially when it comes to deriving computationally efficient recursive algorithms for simulation and inverse dynamics (Luh et al., 1980). Kane (Kane et al., 1983; Kane and Levinson, 1985) presents a detailed treatment of this topic and their approach for eliminating non-contributing forces provides physical insight, is control oriented, and allows for fast simulations. Kane's equations can also be implemented using quasi-coordinates, which we will discuss later.

Also Lagrange's equations of motion can be derived from d'Alembert's principle. Similarly to Kane's approach, Lagrange's equations are derived with d'Alembert as a starting point and the forces of constraint are eliminated from the equations. The elimination of these forces are of vital importance when deriving the dynamics of robot manipulators and allows us to project the equations of motion into the directions for which the system physically allows for displacements. These directions are given by the generalised coordinates, and the Lagrange equations are thus only valid when the state variables can be written in terms of generalised coordinates. We will see that for vehicle-manipulator systems this is unfortunate as the state variables of the vehicle in general cannot be written as generalised coordinates.

The second group consists of systems with nonholonomic constraints. Nonholonomic constraints are not as fundamental as the principle of virtual forces, but they represent a very important group of systems. Nonholonomic constraints are constraints that are not integrable, i.e., they occur when the instantaneous velocities of the system are constrained to a subspace of the configuration space, but the set of reachable configurations are not restricted to the same subspace. The classical example is a wheeled mechanism where the friction of the wheels does not allow the mechanism any instantaneous lateral motion and thus reduces the dimension of the velocity state by one, but at the same time the dimension of the configuration space is not reduced.

The papers included in this thesis do not discuss nonholonomic constraints, except very briefly in Chapter 2. We have chosen, however, to mention nonholonomic constraints and virtual work as they are in many ways related to the next topic. Both for the virtual work approach and nonholonomic forces the state variables need to be projected into the admissible space, and this projection has a clear physical interpretation. Similarly, when the velocity variables cannot be found simply by differentiating the position variables, the state variables also need to be projected into the admissible space so that the topology of the configuration space is maintained. In the sense that some component of the state space, i.e. the velocity, position or forces, needs to be projected to the valid directions,

the three groups discussed here are very much related and a methodology well suited for one type of projection may also be used for another. Kane's equations are for example well suited to incorporate nonholonomic forces in addition to the non-contributing forces (Tanner and Kyriakopoulos, 2001; Lesser, 1992). Kane's equations have not been used for the last group but, as will be clear from Part I, we can reformulate Kane's equations to also include this group.

The third group is thus concerned with projecting from the Euclidean space $\mathbb{R}^n$ onto the configuration space with a different topology. This group differs from the previous two in that the projection is not subject to a physical constraint, but the choice of mathematical representation of the configuration and velocity states. In robotics, most effort regarding this projection deals with projecting onto the Lie groups $SO(3)$ and $SE(3)$. These are important topological spaces in robotics, but unfortunately they are not very well described using generalised coordinates and a vector representation of these spaces leads to singularities. Part I focuses on how we can describe the dynamics of systems with non-Euclidean configuration spaces without the presence of singularities.

1.4.3 Manipulator Dynamics

A wide range of dynamical systems can be described by the Lagrange equations, also commonly referred to as Euler-Lagrange equations (Goldstein et al., 2001)

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}(x,\dot{x})\right) - \frac{\partial \mathcal{L}}{\partial x}(x,\dot{x}) = \tau \tag{1.1}$$

where $x \in \mathbb{R}^n$ is a vector of generalised coordinates, $\tau \in \mathbb{R}^n$ is the vector of generalised forces and

$$\mathcal{L}(x,\dot{x}) \,:\, \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} := \mathcal{U}(x,\dot{x}) - \mathcal{V}(x). \tag{1.2}$$

Here, $\mathcal{U}(x,\dot{x})$ is the kinetic and $\mathcal{V}(x)$ the potential energy functions. We assume that the kinetic energy function is positive definite and in the form

$$\mathcal{U}(x,\dot{x}) := \frac{1}{2}\dot{x}^{\mathsf{T}} M(x)\dot{x}. \tag{1.3}$$

where $M(x)$ is the inertia matrix. For a kinetic energy function in this form we can recast the Lagrange equations (1.1) into the equivalent form

$$M_{RB}(x)\ddot{x} + C_{RB}(x,\dot{x})\dot{x} + n(x) = \tau \tag{1.4}$$

where $C_{RB}(x,\dot{x})$ is the Coriolis and centripetal matrix and $n(x)$ is the potential forces vector defined as

$$n(x) := \frac{\partial \mathcal{V}(x)}{\partial x}. \tag{1.5}$$

Robotic manipulators with conventional revolute or prismatic joints can always be written in terms of generalised coordinates and the Lagrange equations are thus often the preferred choice to derive the manipulator dynamics. This also allows us to find a well defined representation of the Coriolis matrix which has certain useful properties. In

robotics the matrix representing the Coriolis and centripetal forces is normally obtained by the Christoffel symbols of the first kind as (Egeland and Gravdahl, 2003)

$$C_{RB}(x,\dot{x}) := \{c_{ij}\} = \left\{ \sum_{k=1}^{n} \Gamma_{ijk}\dot{x}_k \right\}, \tag{1.6}$$

$$\Gamma_{ijk} := \frac{1}{2}\left(\frac{\partial m_{ij}}{\partial x_k} + \frac{\partial m_{ik}}{\partial x_j} - \frac{\partial m_{kj}}{\partial x_i} \right) \tag{1.7}$$

where $M(x) = \{m_{ij}\}$. When representing the dynamic equations using generalised coordinates we implicitly introduce non-inertial frames in which we represent the inertial properties of the rigid bodies. The Coriolis matrix arises as a result of these non-inertial frames. We note that there are several ways to define the Coriolis matrix so that $c_{ij}(x,\dot{x})\dot{x}_j = \Gamma_{ijk}\dot{x}_j\dot{x}_k$ is satisfied.

Of special interest in this section is the fact that (1.6) and (1.7) require generalised coordinates. For each component we multiply the partial derivative of the inertial matrix with respect to the configuration state x_k with the velocity state $\dot{x}_k$. This is only meaningful if the integral of the velocity state can physically be interpreted as the configuration state. For robotic manipulators this is true, but for most vehicles, as we shall see, this is not the case.

1.4.4 Singularity Prone Dynamics of Single Rigid Bodies

It is a well known fact that the kinematics of a rigid body contains singularities if the Euler angles are used to represent the orientation of the body and the joint topology is not taken into account. One solution to this problem is to use a non-minimal representation such as the unit quaternion to represent the orientation. This will, however, increase the complexity of the implementation as the number of variables is increased by one and this representation cannot be used in Lagrange's equations. This is a major drawback when it comes to modelling vehicle-manipulator systems, mainly because a large class of methods used for robot modelling are based on the Lagrangian approach. It is thus a great advantage if also the vehicle dynamics can be derived from the Lagrange equations.

The problem arises due to the fact that we cannot simply differentiate the position variables to get the velocity variables. This is normally solved by introducing a transformation matrix and we get the relation

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} c\psi c\theta & -s\phi c\theta + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta & 0 & 0 & 0 \\ s\psi c\theta & s\psi c\phi + c\phi s\theta s\psi & -s\psi s\phi + s\theta s\psi c\phi & 0 & 0 & 0 \\ s\theta & c\theta s\phi & c\theta c\phi & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & s\phi t\theta & c\phi t\theta \\ 0 & 0 & 0 & 0 & c\phi & -s\phi \\ 0 & 0 & 0 & 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \tag{1.8}$$

relating the local and global velocity variables. Here $s\theta$ means $\sin\theta$ etc. The well known Euler angle singularity appears as θ approaches $\frac{\pi}{2}$.

Given this relation the dynamics of a single rigid body, such as a ship or an AUV, is normally given by (Fossen, 2002)

$$\dot{\eta} = J(\eta)\nu, \tag{1.9}$$

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + n(\eta) = \tau \tag{1.10}$$

where $\eta = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^\mathsf{T}$ is the position and orientation of the vessel given in the inertial frame, $\nu = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^\mathsf{T}$ is the linear and angular velocities given in the body frame and $J(\eta)$ is given by (1.8). For underwater vehicles $D(\nu)\nu$ is the damping and friction matrix and $M = M_{RB} + M_A$ and $C(\nu) = C_{RB}(\nu) + C_A(\nu)$ represent the rigid body and added mass inertia and the corresponding Coriolis matrices. $n(\eta)$ is the vector of gravitational and buoyancy forces.

The fact that the integral of the local velocity variables has no physical meaning means that the Lagrangian approach cannot be used directly to derive the dynamic equations. This can, however, be circumvented in several different ways. One solution is to introduce a more geometric approach. Geometric methods are in general very convenient when dealing with single rigid bodies, but for multibody systems this approach becomes very complex. We can also introduce so-called quasi-coordinates which are well suited for modelling systems which cannot be expressed in terms of generalised coordinates but most of these methods do not eliminate the singularity from the dynamic equations. In the following we will look briefly at some geometric methods in robotics and the use of quasi-coordinates.

1.4.5 Geometric Methods in Robot Modelling

Lie groups and algebras as a mathematical basis for the derivation of the dynamics of mechanical systems has been used to obtain a singularity free formulation of dynamics (Selig, 2000; Park et al., 1995). We then choose the coordinates generated by the Lie algebra as local Euclidean coordinates which allows us to describe the dynamics locally. Lie groups are manifolds and thus also locally Euclidean. This means that locally we can write the vector of velocity variables as the derivative of the position vector. This is an important property that we can use to write the dynamics of a single rigid body in a singularity free manner. For this approach to be valid globally the total configuration space needs to be covered by an atlas of local exponential coordinate patches. The appropriate equations must then be chosen according to the current configuration. The geometric approach presented in Bullo and Lewis (2004) can then be used to obtain a globally valid set of dynamic equations on a single Lie group, such as an AUV or spacecraft with no robotic manipulator attached. This approach is also used in Marsden and Ratiu (1999).

A mechanical system consisting of joints and links can be modelled as a serial combination of several Lie groups, representing the freedom of each joint. These representations are not only mathematically correct, the singularities are also removed from the equations. Unfortunately, the formulation is more involved than other methods and even though combinations of Lie groups theoretically can be used to represent multibody systems, the formulation is very complex. These approaches are thus not suited for implementation in a simulation or control environment even though, mathematically speaking, they are correct.

The fact that the integral of the local velocity variable has no physical meaning means that the Lagrangian approach cannot be used directly to derive the dynamic equations. For single Lie groups, however, these systems can be modelled using the Euler-Poincaré equations. We can find the Euler-Poincaré equations from Hamilton's extended principle, which states that for two fixed end points, we have (Egeland and Gravdahl, 2003)

$$\int_{t_1}^{t_2} (\delta \mathcal{L} + W_\delta) dt = 0. \tag{1.11}$$

Here, $\delta\mathcal{L}$ denotes the *variation* of the Lagrangian $\mathcal{L}$, which is to be considered a mathematical tool that reflects an infinitesimal change in $\mathcal{L}$ without any change in the physical variable $\mathcal{L}$. Assume that only kinetic energy is present so $\mathcal{L} = \mathcal{U} = \frac{1}{2} V^\mathsf{T} M V$, where $\mathcal{U}$ denotes the kinetic energy and $V = \begin{bmatrix} v & \omega \end{bmatrix}^\mathsf{T}$ is the velocity variable (twist). W_δ is the virtual work of the active generalised forces and is given by

$$W_\delta = \sum_{j=1}^{n} \tau_j \delta q_j. \tag{1.12}$$

We can now derive the attitude dynamics of a rigid body on $SO(3)$, i.e. $V = \omega$, with only kinetic energy. We write $\dot{R} = R\hat{\omega}$ for a rotation matrix R and the angular body velocities ω, and $\delta R = R\hat{\sigma}$ for an arbitrary vector σ, where $\hat{p} \in \mathbb{R}^{3\times 3}$ is the skew-symmetric matrix such that $\hat{p}x = p \times x$ for all $p, x \in \mathbb{R}^3$. Hamilton's extended principle gives (Egeland and Gravdahl, 2003)

$$\begin{aligned} \int_{t_1}^{t_2} (\delta \mathcal{L} + W_\delta) dt &= \int_{t_1}^{t_2} \left(\frac{\partial \mathcal{U}}{\partial \omega} \delta\omega + \tau_\omega^\mathsf{T} \sigma \right) dt \\ &= \int_{t_1}^{t_2} \left(\frac{\partial \mathcal{U}}{\partial \omega} (\dot{\sigma} + \hat{\omega}\sigma) + \tau_\omega^\mathsf{T} \sigma \right) dt \\ &= \int_{t_1}^{t_2} \left(-\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial \mathcal{U}}{\partial \omega} \right) + \frac{\partial \mathcal{U}}{\partial \omega} \hat{\omega} + \tau_\omega^\mathsf{T} \right) \sigma dt \end{aligned} \tag{1.13}$$

The Euler-Poincaré equations of motion on $SO(3)$ are then given from (1.11) and (1.13) as

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial \mathcal{U}}{\partial \omega} \right)^\mathsf{T} + \hat{\omega} \left(\frac{\partial \mathcal{U}}{\partial \omega} \right)^\mathsf{T} = \tau_\omega \tag{1.14}$$

and similarly the Euler-Poincaré equations of motion on $SE(3)$ are written as

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial \mathcal{U}}{\partial V} \right)^\mathsf{T} - \mathrm{ad}_V^\mathsf{T} \left(\frac{\partial \mathcal{U}}{\partial V} \right)^\mathsf{T} = \tau \tag{1.15}$$

where $\mathrm{ad}_V = \begin{bmatrix} \hat{\omega} & \hat{v} \\ 0 & \hat{\omega} \end{bmatrix}$.

If we write $\tau = \begin{bmatrix} \tau_1^\mathsf{T} & \tau_2^\mathsf{T} \end{bmatrix}^\mathsf{T}$ we get the well known Kirchhoff's equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{U}}{\partial v}\right)^\mathsf{T} + \hat{\omega}\left(\frac{\partial \mathcal{U}}{\partial v}\right)^\mathsf{T} = \tau_1 \tag{1.16}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{U}}{\partial \omega}\right)^\mathsf{T} + \hat{v}\left(\frac{\partial \mathcal{U}}{\partial v}\right)^\mathsf{T} + \hat{\omega}\left(\frac{\partial \mathcal{U}}{\partial \omega}\right)^\mathsf{T} = \tau_2 \tag{1.17}$$

Kirchhoff's equations are widely used to model ships and underwater marine vessels as well as spacecraft. We note however, that the Euler-Poincaré equations of motion on both $SO(3)$ and $SE(3)$ assume kinetic energy only and do not depend on the position variables. The Euler-Poincaré equations use only the local velocity variables and we thus avoid the transformation between the local and global velocity variables. As the position variables are the integrated *global* velocity variables, we note that we need this transformation to include the position variables in the formulation. We thus obtain a singularity free formulation at the expense of the potential energy and position variables.

The reason that we can obtain this singularity free representation is basically that we are dealing with single bodies, and that the inertia matrix is constant and the kinetic energy is thus given by $\mathcal{U}(\omega) = \frac{1}{2}\omega^\mathsf{T} M \omega$, for $SO(3)$ and similarly for other configuration spaces. For multibody systems the inertia matrix depends on the configuration of the system, and so does forces such as gravity, which are also configuration dependent. In this sense the problem of obtaining the position variables from the local velocity variables is not solved.

There is one specific multibody system where the formulation above can be applied. Egeland and Pettersen (1998) derive the singularity free dynamic equations of spacecraft-manipulator systems. These systems are special in two ways which makes it possible to expand Kirchhoff's equations to multibody systems. Firstly, there are no gravitational forces present and the orientation of the spacecraft is not needed in the equations. Secondly, the orientation of the spacecraft is not needed in the inertia matrix which only depends on the position of the robot joints, which are Euclidean and thus not subject to singularities. However, if the attitude of the spacecraft needs to be determined, the transformation between the local velocity variables and the position variables is once again required.

1.4.6 Quasi-Coordinates and Quasi-Velocities

Another way to deal with the fact that the integral of the velocity variable has no physical meaning is to introduce quasi-coordinates (Gingsberg, 2007). In many cases it is easier to formulate the dynamic equations in terms of velocity variables that cannot simply be written as the time derivative of the position variables. For example, when dealing with angular motion the angular velocity is not the rate of change at which a rotation angle changes, except for the planar case. There is thus no finite change in orientation that corresponds to the angular velocity. We will denote such velocity variables quasi-velocities, often represented by $\dot{\gamma}$. The corresponding quasi-coordinate γ does not have any physical interpretation itself whereas $\dot{\gamma}$ has the physical interpretation of $\mathrm{d}\gamma = \dot{\gamma}\mathrm{d}t$, i.e. it is associated with differential increments.

Quasi-coordinates are hence velocity coordinates that are not simply the time derivative of the position coordinates. The relation between the quasi-velocities v and the derivative of the configuration variables q (generalised coordinates) is given by

$$v = S(q)\dot{q}. \tag{1.18}$$

There are many ways to write the dynamics in terms of quasi-coordinates. The Gibbs-Appel equations (Gibbs, 1879; Lewis, 1996) are obtained by differentiation of a function of kinematic variables. In this way the Gibbs-Appel equations resemble Lagrange's equations and the Euler-Poincaré equations. On the other hand, Kane's equations (Kane et al., 1983; Kane and Levinson, 1985) use the virtual work principle as a starting point and can be considered a generalisation of the approaches that are derived from the virtual work approach.

There are also some more recent advances. Quasi-coordinates are used in Kwatny and Blankenship (2000) where the Poincaré equations of motion are derived in terms of quasi-velocities. Let X be defined in terms of the Lie bracket as

$$X_j = \big[\big[t_j, t_1\big]\,\big[t_j, t_2\big] \cdots \big[t_j, t_m\big]\big] \tag{1.19}$$

where $T = \begin{bmatrix} t_1 & t_2 & \cdots & t_m \end{bmatrix}$ is given below in (1.20) and the Lie bracket is defined as $\big[t_1, t_2\big] = t_1 t_2 - t_2 t_1$. The Poincaré equations of motion are then given by

$$v = S(q)\dot{q}, \qquad \dot{q} = T(q)v \tag{1.20}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial v}\right) - \frac{\partial \mathcal{L}}{\partial q}T - \sum_{j=1}^{m} v_j \frac{\partial \mathcal{L}}{\partial v} S X_j = \tau T \tag{1.21}$$

for the Lagrangian $\mathcal{L}(q, v)$ in terms of quasi-velocities. If we substitute (1.20) into (1.21) we get

$$\dot{v}\frac{\partial^2 \mathcal{L}}{\partial v^2} + vT\frac{\partial^2 \mathcal{L}}{\partial q \partial v} - \frac{\partial \mathcal{L}}{\partial q}T - \sum_{j=1}^{m} v_j \frac{\partial \mathcal{L}}{\partial v} S X_j = \tau T. \tag{1.22}$$

We note that the configuration coordinates q are present in the equations and that it is assumed that $q \in \mathbb{R}^n$. In other words the relation between the derivatives of the position variables and the quasi-velocities depends on the vector q, which we have already seen leads to singularities in the representation. We see this by writing out the expression for $T(q)$ as found in Kwatny and Blankenship (2000) for the $SE(3)$ case:

$$T(q) = \begin{bmatrix} * & * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{s\phi s\theta}{c\theta} & \frac{c\phi s\theta}{c\theta} \\ 0 & 0 & 0 & 0 & c\phi & -s\phi \\ 0 & 0 & 0 & 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \tag{1.23}$$

which is singular at $\theta = \frac{\pi}{2}$.

Once the relation in (1.18) is established, the dynamics can also be written in the form of the Boltzmann-Hamel equations. The Boltzmann-Hamel equations are derived from the Lagrangian, but allows the velocity variable to be written in the form of (1.18) as (Duindam and Stramigioli, 2007, 2008)

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial v}\right) - S^{-\mathsf{T}}\frac{\partial \mathcal{L}}{\partial q} + \left(\sum_k \gamma_k v_k\right)\frac{\partial \mathcal{L}}{\partial v} = \tau \tag{1.24}$$

with the Lagrangian

$$\mathcal{L}(q,v) = \mathcal{U}(q,v) - \mathcal{V}(q) = \frac{1}{2}v^{\mathsf{T}}M(q)v - \mathcal{V}(q) \tag{1.25}$$

written in terms of the quasi-velocities and quasi-coordinates and

$$(\gamma_k)_{ij}(q) = \sum_{l,m} S^{-1}_{li} S^{-1}_{mk}\left(\frac{\partial S_{lj}}{\partial q_m} - \frac{\partial S_{lm}}{\partial q_j}\right). \tag{1.26}$$

The Boltzmann-Hamel equations thus differ from the Poincaré equations in how the derivatives are computed. While the Poincaré equations use the Lie bracket, the Boltzmann-Hamel equations require the partial derivatives of the transformation (1.18). The Boltzmann-Hamel equations are also used in Cameron and Book (1997) and Jarzbowska (2008), and in Maruskin and Bloch (2007) the same equations are used to solve the optimal control problem for nonholonomic systems. Nonholonomic constraints are easily included in the dynamics using the Boltzmann-Hamel equations, and quasi-coordinates in general. In Kozlowski and Herman (2008) and Herman and Kozlowski (2006) several control laws using a quasi-coordinate approach were presented.

Common for all these methods is, however, that the configuration space of the vehicle and robot is described as $q \in \mathbb{R}^n$. This is not a problem when dealing with 1-DoF revolute or prismatic joints but for more complicated joints such as ball-joints or free-floating joints this does not solve the singularity problem, as illustrated by (1.23). Joints with more than one degree of freedom are sometimes modelled as compound kinematic joints (Kwatny and Blankenship, 2000), i.e., a combination of 1-DoF simple kinematic joints. For joints that use the Euler angles to represent the generalised coordinates also this solution leads to singularities in the representation.

1.4.7 Advantages of the Proposed Approach

From the previous sections we can divide the approaches that deal with non-Euclidean transformations into two groups:

- The geometric approach - allows us to represent transformations with a different topology than that of $\mathbb{R}^n$ globally, but is not suited for multibody systems.
- The quasi-coordinate approach - allows us to project the velocities into the directions allowed by the kinematics constraints, and more importantly, project the velocities so that the configuration space is consistent with the manifold. These approaches handle multibody systems very well, but singularities arise in the transformation between the local and global velocity variables.

In this section we close the gap between the two approaches and find the dynamic equations of vehicle-manipulator systems using quasi-coordinates, but without the singularities that normally arise (Kwatny and Blankenship, 2000; Cameron and Book, 1997; Jarzbowska, 2008; Maruskin and Bloch, 2007; Gibbs, 1879; Lewis, 1996; Kane et al., 1983; Kane and Levinson, 1985). The approach is based on Duindam and Stramigioli (2007, 2008) and we extend these ideas to vehicle-manipulator systems. We also include external forces and considerations that do not normally appear in fixed-base manipulators. Before we go into details on the proposed approach, we list the main advantages:

- The approach is globally valid and there are thus no singularities in the representation.
- The approach is general and allows for joints with configuration spaces different than that of $\mathbb{R}^n$ anywhere in the chain.
- The approach allows us to use results from the Lie theory, but is valid also for multibody systems and thus an extension of the approaches that are only valid for single Lie groups.
- Several results from the Lie theory allow for easy and computationally efficient implementation of the final equations.
- No local state variables are present in the final equation.
- The use of quasi-coordinates allows us to write the kinematics in a block-diagonal form and the kinematics of each joint can thus be specified at joint level.
- Specifying the kinematics at joint level allows us to describe systems with more than one non-Euclidean joint.
- The block-diagonal form allows us to build up a library of different joints and is thus suited for software implementation.
- The approach is well suited for vehicle-manipulator systems as it allows us to include gravitational forces, buoyancy forces and the configuration of the vehicle.

We note that none of the approaches mentioned in the previous sections possess all these properties.

We follow the generalised Lagrangian approach presented in Duindam and Stramigioli (2007, 2008) which allows us to combine Euclidean joints and more general joints, i.e., joints that can be described by the Lie group $SE(3)$ or one of its ten subgroups, and we extend these ideas to vehicle-manipulator systems. As we have seen, there are several advantages of following this approach. The use of quasi-coordinates allows us to include joints (or transformations) with a different topology than that of $\mathbb{R}^n$. For example, for an AUV we can represent the transformation from the inertial frame to the AUV body frame as a free-floating joint with configuration space $SE(3)$ and we avoid the singularity-prone kinematic relations between the inertial frame and the body frame velocities that normally arise in deriving the AUV dynamics (Fossen, 2002). This transformation is subject to the well known Euler angle singularities and the dynamics are not valid globally.

We have seen that there are several different ways to derive the dynamic equations in terms of quasi-coordinates. Common for all these approaches, however, is that the transformation matrix in (1.18) is not globally defined. The reason for this is that the position variables are assumed to be of the form $q \in \mathbb{R}^n$. With this as a starting point we are not able to find a well defined matrix $S(q)$ for which (1.18) is true.

Instead we represent the configuration of the manipulator as a set of configuration states $Q = \{Q_i\}$. The configuration state Q_i of joint i is then the matrix representation of the Lie group corresponding to the topology of the joint. The corresponding block $S_i(Q_i)$ relating the local and global velocity variables is well known from the Lie theory and can be found in terms of the Lie bracket or the exponential map (Rossmann, 2002). Note that S_i depends only on Q_i. For standard revolute and prismatic joints Q_i becomes a scalar $Q_i = q_i$ and $S_i = 1$.

The main tool that we use is the exponential map. This allows us to express the dynamics in exponential coordinates ϕ and locally every state Q_i is described by a set of Euclidean coordinates $\phi_i \in \mathbb{R}^{n_i}$. Thus, in the neighbourhood of $\bar{Q}_i$ there exist a function $\Phi_i(\bar{Q}_i, \phi_i)$ that defines a local diffeomorphism between a neighbourhood of $0 \in \mathbb{R}^{n_i}$ and a neighbourhood of $\bar{Q}_i$. $\bar{Q}_i$ is thus locally described by $Q_i = \Phi_i(\bar{Q}_i, \phi_i)$ with $\Phi_i(\bar{Q}_i, 0) = \bar{Q}_i$.

The allowed joint velocity is given as an element of the tangent space of the Lie group. This is uniquely described by a vector $v_i \in \mathbb{R}^{n_i}$. For Euclidean 1-DoF joints we thus get $v = \dot{Q} \in \mathbb{R}$ and the coordinate mapping is given by $\Phi(\bar{Q}, \phi) = Q + \phi$ with $\phi \in \mathbb{R}$. The most important group in this thesis is joints with a Lie group topology. Φ is then given by the exponential map, i.e.

$$\Phi(\bar{Q}, \phi) = \bar{Q} e^{\sum_{i=1}^{n_i} b_i \phi_i} \tag{1.27}$$

where b_i is the basis elements of the Lie algebra. The block $S_i(Q_i)$ is now found by

$$\hat{v} = \dot{\Phi}(\bar{Q}, \phi)\Phi^{\mathsf{T}}(\bar{Q}, \phi). \tag{1.28}$$

Here we can factorise $\dot{\phi}$ and find the relation between v and $\dot{\phi}$. For example, for $SE(3)$ (1.28) becomes

$$v = \left(I - \frac{1}{2} \operatorname{ad}_\phi + \frac{1}{6} \operatorname{ad}_\phi^2 - \cdots \right) \dot{\phi}. \tag{1.29}$$

Another important observation at this stage is that if we think of the global position variable Q as a parameter (not as a variable), and the local position variable ϕ as a state variable, we can write the Lagrangian in terms of this position variable and the velocity state v, i.e. $\mathcal{L}(\phi, v)$. The dynamic equations can now be found using the Boltzmann-Hamel equations. Now, if we evaluate the final equations at $\phi = 0$, we get back our global state variables as $\Phi_i(Q_i, 0) = Q_i$ and the dynamics are written in terms of Q and v.

This has two main advantages. Firstly, the dynamics do not depend on the local coordinates as these are eliminated from the equations and the global position and velocity coordinates are the only state variables. This makes the equations valid globally. Secondly, evaluating the equations at $\phi = 0$ greatly simplifies the dynamics and makes the equations suited for implementation in simulation software. This is seen easily in (1.29) where all the higher order terms vanish and after differentiating and setting $\phi = 0$ only the second term in the parenthesis is non-zero. We also note that the approach is well suited for model-based control as the equations are explicit and without constraints.

Even though the expressions in the derivation of the dynamics are somewhat complex the final expressions can be written in a very simple form. We present several examples of how we can use well known results—in addition to some new relations—to simplify the dynamic equations and speed up the implementation.

The main purpose of Part I is to study systems that consist of a moving vehicle with a robotic manipulator attached to it. To our best knowledge these systems have not been studied in detail in literature using the framework presented here. Nor are we aware of other formulations of vehicle-manipulator systems with all the properties listed above. The use of quasi-coordinates to derive the dynamics in this way has mainly been applied to standard robotic manipulators with 1-DoF joints with the extension to more general types of joints in Kwatny and Blankenship (2000) and Duindam and Stramigioli (2008). However, the treatment of vehicle-manipulator systems deserves a special treatment. Chapter 2 (From et al., 2010a) presents a detailed study of vehicle-manipulator systems and also shows how the dynamics can be implemented in a computationally efficient manner. Chapter 6 (From et al., 2010e) presents a detailed study of the dynamically equivalent manipulator (DEM) approach for space manipulators and we present the singularity free dynamic equations of the DEM for the first time. In Chapter 5 we study some important properties of these systems, namely the boundedness property of the inertia matrix and the skew-symmetric property of the Coriolis matrix.

1.4.8 Motion Planning and Control of Robots a Non-Inertial Base

From the previous sections we see that we can derive the dynamics of vehicle-manipulator systems in a robust and computationally efficient way. The explicit equations presented serve as a good platform for gaining a deeper insight into the dynamic coupling between the moving base and the manipulator. Based on the formulation presented in Chapter 2 (From et al., 2010a) we develop several new algorithms for motion planning and control of a robot mounted on a moving base. First, in Chapter 3 (From et al., 2009a) we study how the non-inertial forces of a moving ship affect a manipulator mounted on the ship and we show that if we take the non-inertial forces into account in the motion planning and control of the manipulator we can substantially reduce the torques needed to take the manipulator from an initial to a final configuration. Then, in Chapter 4 (From et al., 2009b) we run several experiments and show that even though the future ship motion can only be predicted with a certain accuracy, we can include the predicted ship motion in the motion planner and improve performance. We also study in detail for the first time how the motion of the non-inertial base affects the manipulator and show through experiments that a manipulator mounted on a ship in only 1 meter wave height has to use a substantial amount of torque just to compensate the inertial forces. This result serves as a good motivation for the papers in this section and illustrates the importance of these studies.

1.5 Part II - Robust Manipulator Design

As already discussed in the previous section, robustness is an extremely critical issue when it comes to autonomous operation of remotely located facilities. In Part II of this thesis we

are not only concerned with the mathematical representation of the robots for simulation and control, but also with the mechanical design of the robot. The main topic covered in this part is fault tolerance with respect to torque failure, or free-swinging joint faults.

Both serial and parallel robots are widely used in remote and harsh environments where humans cannot or do not want to operate. In the setting of joint failure there is an important difference between serial and parallel manipulators. We find that for serial manipulators the fault tolerance problem needs to be addressed in the control of the robot after joint failure occurs, while for parallel manipulators this needs to be addressed in the mechanism design. In any case, the need for a rigorous theory on what happens when joint failure occurs is important to be able to cope with unforeseen events of this kind. In this part we present a complete and mathematically rigorous theory of the effects that passive joints have on serial and parallel manipulators when external forces are present.

Even though the theory developed in this part is presented in the setting of joint failure, it also gives us valuable insight into how passive joints affect serial and parallel mechanisms in general. The proposed framework is also well suited for mechanism design of parallel manipulators and motion analysis of such systems.

1.5.1 Joint Failure

Fault tolerance needs to be handled differently for different types of joint failure. The most important types of joint failure are:

- Free-swinging joint faults (FSJF) - this is the case when an actuated joint turns passive because, for some reason, no torque is available to control the joint. This is also known as torque failure.
- Locked joint faults (LJF) - this is the case when the joint, for some reason, becomes stiff, or locked.
- Incorrectly measured joint position or velocity faults (JPF/JVF) - this occurs when the measurements of the joint position or velocity becomes unavailable or are measured incorrectly.

We are only concerned with free-swinging joint failure. Electrical motors can fail, and when they do it is important to reduce the damage caused by this failure. Due to external forces such as gravity joint failure of this type may cause the mechanism to collapse. However, this can sometimes be avoided if the correct precautions are taken in the design or the control of the mechanism. This is the main topic of this part.

Figures 1.8 and 1.9 illustrate the difference between serial and parallel mechanisms. In Figure 1.8 we see that a passive joint at the end of a serial chain leads to a motion that cannot be compensated for if an arbitrary external force is applied. The beam may thus cause severe damage to the surroundings.

Closed chain manipulators such as parallel manipulators and cooperating serial manipulators have many advantages over their serial counterparts. Parallel manipulators are stiffer, faster, and more accurate than serial manipulators at the cost of a smaller workspace. Cooperating robots can handle heavier and larger objects than serial manipulators and are thus the preferred choice in many applications. Closed chain mechanisms are also more resistant to torque failures. From Figures 1.8 and 1.9 it is clear that a parallel

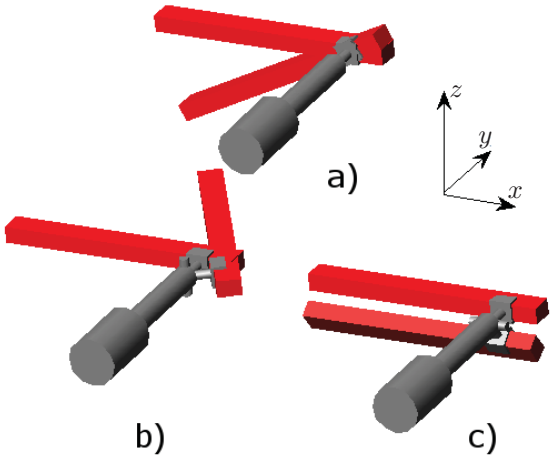


Figure 1.8: Passive joint about the a) y-, b) z- and c) x-axis for a serial manipulator.

manipulator will deal much better with a FSJF than a serial manipulator. For example, the FSJF in Figure 1.9c) will not be affected by the external forces at all.

1.5.2 Serial Manipulators

For serial manipulators joint failure is extremely serious and will always result in an undesired motion if an arbitrary external force is present. Also, this undesired motion cannot be compensated for by controlling the active joints. We will denote a motion generated by the passive joints a *passive motion*. Because serial manipulators do not have any closed loops, and thus no loop constraints in the mechanism, there will always exist an external force for which such a passive motion is generated. For example, if the passive motion is affected by the gravitational forces, the mechanism, or part of the mechanism, will fall to the ground.

When the external force is known we can in some cases prevent the manipulator from collapsing and protect the surroundings of the manipulator. In this case we investigate under what conditions, i.e. for what external forces and for what configurations of the robot, the external forces do not affect the motion of the passive joints. We will say that the manipulator is conditionally equilibrated with respect to an external force at all configurations for which the passive joints are not affected by this specific external force. Thus, if joint failure occurs we can prevent damage to the robot's surroundings by taking the manipulator to one of these configurations in a controlled manner. For example, if the only external force is a force in the direction of the z-axis (e.g. gravity), the joint in Figure 1.8b) will resist this force as opposed to 1.8a) or 1.8c) and we should thus strive to reach such a configuration.

From this it is clear that joint failure is extremely serious for serial manipulators. However, the solution proposed in this part allows us to make the best out of a very difficult and dangerous situation and may prevent serious damage to the robot's surroundings.

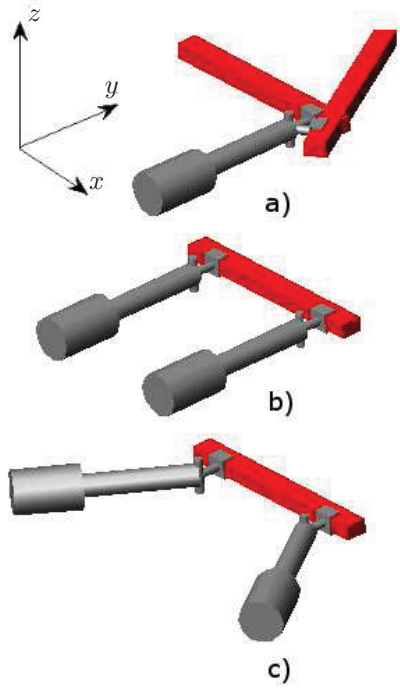


Figure 1.9: The conceptual differences of passive joints in serial and parallel manipulators.

1.5.3 Parallel Manipulators

For serial manipulators the fault tolerance needs to be addressed in the control of the robot after such a failure has been identified. For parallel manipulators, however, the problem of free-swinging joint failure can—and should—be addressed in the design of the robot. Parallel mechanisms may have passive joints as an intrinsic property, i.e. a design choice, or due to joint failure. In any case, it is important to understand how the passive joints affect the mobility and controllability of the robot.

Because of the closed chains, a joint failure may not affect the mechanism at all. In the papers presented in Section II we will see several examples of robots that do not collapse, or can even continue operation, when joint failure occurs. To be able to continue operation, we need to choose one or more redundant actuated joints and we study in detail how to choose the active and passive joints of the mechanism to guarantee fault tolerance using as few actuated joints as possible.

We thus find that parallel manipulators can be designed such that all the degrees of freedom of the motion remain controllable when joint failure occurs for an arbitrary joint. This will, however, require more active joints than necessary to control the degrees of freedom the manipulator originally was intended to operate in. This actuator redundancy is in many cases undesirable due to manufacturing and maintenance costs, weight, performance, and so on. The number of redundant active joints should thus be kept as low as possible. We present a mathematically rigorous theory for identifying the smallest set of

active joints for which the mechanism is fault tolerant.

If the fault tolerance problem is not addressed in the design process it must be handled in the control of the manipulator in the case of such an occurrence. However, we find that this is more difficult for parallel than for serial manipulators and the problem of joint faults should be addressed in the design process to guarantee a fault tolerant solution.

1.5.4 Relevant Literature and Advantages of the Proposed Approach

For parallel manipulators we first need to investigate if a joint failure will allow a passive motion or not. The relation between the active and passive joints in a parallel manipulator is important in this setting and well covered in literature. The Jacobian of the parallel manipulator is investigated in Liu et al. (1999) and Bicchi and Prattichizzo (2000) where the passive joint accelerations are found from the active joint accelerations by dividing the Jacobian into an active and a passive part. This tells us how the passive joints move due to the controlled motion of the active joints, but it does not give us the motion of the different links directly. The motion of the passive joints is then given in the form (Liu et al., 1999)

$$\dot{\theta}_p = -J_p^{\dagger}(\theta) J_a(\theta) \dot{\theta}_a \tag{1.30}$$

where $J_p^{\dagger}$ is the pseudo-inverse of the passive Jacobian and J_a is the active Jacobian.

We will use the concept of mobility which has been studied by various researchers. The mobility basically tells us how many degrees of freedom the mechanism has, counting both the motion of the end effector and the internal motion of each chain. For non-overconstrained mechanisms, i.e. when there are no redundant constraints, we can find the mobility by the well known Grübler formula (Murray et al., 1994). For overconstrained mechanisms there are many approaches to determine the mobility. The two most important approaches in this setting are the constraint space and the motion space approaches.

In Dai et al. (2006) the mobility of the mechanism is found from the constraint space. The constraints of the system are found systematically and the redundant constraints are identified. The mobility of the mechanism is then found by adding the degrees of freedom represented by these redundant constraints to the Grübler formula for non-overconstrained mechanisms. This approach illustrates well the effects of redundant constraints in the mechanism.

The mobility can also be found by the motion space as in Rico et al. (2003, 2006). The degree of freedom of the motion of the end effector is first found. Then the degree of freedom of the self-motion manifold of each chain is added. By this approach the redundant constraints are not found directly, but this approach gives valuable insight into where to place redundant actuators in the mechanism, which is our main objective.

For serial manipulators, passive joints are not an intrinsic property of the manipulator and this has thus not been treated to any extent in literature. A few exceptions can be found, though, for example in the study of the inverted pendulum with a passive joint. Some references that address passive joints in serial manipulators are Oriolo and Nakamura (1991) and Arai and Tachi (1990), and case studies such as the Acrobot (Hauser and Murray, 1990).

For a comprehensive treatment on how to identify joint failures see Tinós et al. (2007). Once these are identified the appropriate control actions should be applied to minimise

damage to the surroundings. Fault tolerant actuators have recently been proposed as another way to deal with fault tolerance with respect to torque failure (Muenchhof et al., 2009; Steffen et al., 2009). The main idea here is to use several small actuators and thus guarantee that if one actuator fails, the remaining actuators can still apply the necessary torque. This approach will not, however, be tolerant with respect to problems such as power failure.

The approach presented here is based on the motion space approach, as presented in Rico et al. (2003, 2006), and we verify if the manipulator—considering the passive joints only—generates a non-trivial motion. By considering the passive joints only, and assuming all actuated joints as fixed, we gain valuable insight into the uncontrollable motion of the passive joints in the mechanism. Then, if the passive joints of the manipulator allow a motion, we investigate what kind of motion it implements. From this we can conclude; (i) given a mechanism, with respect to what kind of external forces is the manipulator equilibrated; and (ii) given an external force, what kind of mechanism and for what configurations is the mechanism equilibrated with respect to the external force.

The mobility and classification of robotic mechanisms can be based on several different frameworks. Lie groups have been extensively used as a basis to represent the motion of the joints, the chains and the end effector. The early work of Reuleaux (1875) recognised some commonly used primitive generators as Lie subgroups. The most extensive work on mechanism topology in this setting is maybe the work by Hervé (1978), Hervé and Sparacino (1991) and Lee and Hervé (2006) where the topology of parallel mechanisms was studied based on a Lie geometrical formulation. Based on this work, Meng et al. (2007) developed a geometric theory for a more precise and complete treatment of the synthesis problem. Other related work are Angeles (2004), Li et al. (2004) and Selig (2000).

We use the framework of Meng et al. (2007), where a precise geometric theory for analysis and synthesis of sub-6 DoF manipulators is presented. The low dimensional subgroups or submanifolds of $SE(3)$ are used to represent the lower pairs, or primitive generators (commonly referred to as joints), while the high dimensional subgroups are used to represent the desired end-effector motion types. Given a desired end-effector motion type as a Lie subgroup or a submanifold, the synthesis problem is solved for serial and parallel manipulators, i.e. for a set of primitive generators they find the type of motion that these generators implement. Then, from a pre-specified list of primitive generators, all possible serial and parallel arrangements of the primitive generators so that the resulting manipulator has the desired end-effector motion type are found.

The main contribution of Part II is that we extend this work and present a complete and rigorous theory of the effect that passive joints have on robotic mechanisms. First we use the concept of mobility to define the strongest type of robustness we can obtain for a parallel manipulator:

Definition 1.5. A parallel manipulator $\mathcal{M}$ is denoted *equilibrated* if $\mathcal{M}$, either through kinematic constraints or through actuator torque, can resist an arbitrary external wrench $F_{ext} = \begin{bmatrix} f^{\mathsf{T}} & \tau^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$ where $f, \tau \in \mathbb{R}^3$. In the case that an arbitrary wrench can be accommodated by the kinematic constraints, we will say that the manipulator is *passively sustained*. When an arbitrary wrench can be produced by the actuation torque, we will denote it *actively equilibrated*.

Secondly, we extend the formalism presented in Meng et al. (2007) to analyse the passive motion of a mechanism. Such a passive motion is always possible for serial manipulators with passive joints, or for parallel manipulators that are not found to be equilibrated by Definition 1.5. Using this formalism it is found that a mechanism is conditionally equilibrated with respect to an external force if the mechanism considering only the passive joints is a motion generator of a motion for which the reciprocal product with the external force vanishes. If the passive joints of the manipulator allow a motion, we are interested in what kind of motion this is. This is the concept of conditionally equilibrated mechanism introduced for the first time in the papers in Part II. We will denote a mechanism-force pair conditionally equilibrated if the following holds:

Definition 1.6. A manipulator $\mathcal{M}$ is denoted *conditionally equilibrated* with respect to a given external wrench $F_{ext} = \begin{bmatrix} f^{\mathsf{T}} & \tau^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$ where $f, \tau \in \mathbb{R}^3$, if $\mathcal{M}$, either through kinematic constraints or through actuator torque, can produce a wrench opposite to F_{ext}, i.e. $\mathcal{M}$ can produce the wrench $-kF_{ext}$ for some $k > 0$.

Thus, an equilibrated mechanism can resist any external force, while a conditionally equilibrated mechanism can resist one specific external force, and only for certain configurations. We note that equilibrated mechanism applies to parallel mechanism only, but conditionally equilibrated applies to both serial manipulators and parallel manipulators with passive joints.

The proposed approach is a systematic and rigorous analysis of the mobility of closed chain mechanisms based on the theory of twists. The analysis makes it possible to calculate the mobility of the mechanism based on the number and type of joints in each sub-chain. We then determine the minimum set of active joints needed for the manipulator to be equilibrated and fault tolerant. The mechanism needs to be equilibrated not only with respect to forces acting on the end effector, but also with respect to forces acting on the chains. Thus, in addition to the end-effector motion we also need to consider the internal motion of each chain to guarantee that the mechanism is equilibrated.

We present several examples of how to apply the theory presented to different mechanisms. For three types of mechanisms—exceptional linkages and trivial linkages of type I and II—we show how to choose the minimum number of active joints so that the mechanisms are equilibrated and fault tolerant.

Even though the mobility of closed chain manipulators is given a lot of attention in literature, there does not seem to be a thorough treatment of mobility in the light of joint failure. Torque failure, or free-swinging joint faults are treated in Matone and Roth (1999), English and Maciejewski (1998), Tinós and Terra (2002) and Tinós et al. (2007). The passive motion that these joint failures allow have, however, not been treated in literature.

1.6 Part III - Functional Redundancy

Autonomous operation of remotely located fields depends on robots that can perform a wide variety of tasks. This may be pre-programmed routine operations or tasks that need to be performed in real time. In any case, robotic manipulators that are to perform a wide variety of tasks need to be designed so that the operational space of the manipulator is equal to or bigger than the task space of all the tasks it is to perform. If the dimension of

the task space is smaller than the operational space for a specific task, we potentially have more controllable degrees of freedom for the end effector than necessary to perform the given task. This freedom is known as functional redundancy and can be used to improve performance if dealt with properly in the motion planning and control of the robot.

We are especially interested in this type of redundancy as it will typically arise in settings where one manipulator is to perform several different tasks. In remote areas the robotic manipulator should be able to perform all the tasks previously performed by humans. Such a wide variety of tasks will lead to tasks with different dimensions of the task spaces. Several different tasks with different task dimension will itself lead to functional redundancy.

One common type of application where functional redundancy is utilised to improve performance is the so-called pointing task. In pointing task applications we only need to specify the direction of the end effector and not the orientation. In addition to the three degrees of freedom of the position, the direction of the end effector has only two degrees of freedom, as opposed to the three degrees of freedom of the orientation. For standard industrial manipulators with six degrees of mobility this gives one degree of freedom that can be used to lower energy consumption or the strain and tension on the robots, increase the speed of the job, or improve the quality of the job in general.

We note that this freedom arises due to the task and not the design of the manipulator itself. It is thus fundamentally different from the redundancy that arises in manipulators with many joints in the sense that the freedom arises without the need to add redundant joints to the chain. We can thus obtain functional redundancy without having to increase the production cost and the complexity of the manipulator.

We will introduce a wider definition of functional redundancy than the one normally found in the robotics literature. Functional redundancy is normally defined as an operational space and a task (and thus independent of the manipulator design) where one or more degrees of freedom of the operational space are not of concern for the specific task. For the pointing task the rotation about the end-effector central axis can be chosen arbitrarily and thus represents a one degree of freedom redundancy.

In many applications, however, we may have a freedom in the specifications of the end effector that cannot be described by a complete one degree of freedom redundancy as we could for the pointing task. An easy case to visualise is when we allow a small error in the orientation of the end effector. If the end-effector orientation is specified, but in addition we allow an orientation error around one or more axes, this cannot be described using the conventional framework of functional redundancy. We thus propose a wider definition of functional redundancy which also allows us to describe these cases.

We start by defining the different classifications of redundancy as they are normally defined in literature and the spaces used to describe these. We then present a new, wider interpretation of functional redundancy well suited to represent the kind of freedom that arises in the cases described above.

1.6.1 Joint Space, Operational Space and Task Space

The concept of redundancy can be defined as an intrinsic kinematic property of the manipulator or as a property relative to the task that the manipulator performs. To get a formal definition of redundancy and understanding this subtle difference, we need to define the

different spaces of the manipulator. We use the definitions of Sciavicco and Siciliano (2005).

Firstly, the *operational space* is the space in which the manipulator task is specified. The dimension of this space is given by the minimum number of independent parameters needed to represent this space. In $SE(3)$ we need three parameters to describe the position of the end effector and three for the orientation. The dimension of the operational space in this case is thus $m = 6$. We denote an element of this space by $g \in SE(3)$ and the minimum number of independent parameters needed to represent the matrix g is normally given by the vector representation $g^\vee$, i.e.

$$g^\vee = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix}. \tag{1.31}$$

where x, y, and z determine the position and ϕ, θ, and ψ are the Euler angles. Similarly, for the planar case $SE(2)$ we have $m = 3$.

Secondly, the *joint space*, or configuration space, defines the space on which the joint variables live. This determines what is often referred to as the *degree of mobility* for serial manipulators. If the manipulator consists of 1-dimensional rotational or translational joints only, the dimension of the joint space is equal to the number of joints in the open chain. An element of the joint space is given by a vector

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix}. \tag{1.32}$$

For a standard industrial manipulator we have $n = 6$.

Finally, we define the *task space* as the subspace of the operational space needed to describe a specific task. For example the pointing task can be completely described with only five of the components of the operational space in $SE(3)$, i.e.

$$g_{pt}^\vee = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \end{bmatrix} \tag{1.33}$$

and the dimension of this space is $r = 5$. ψ can thus be chosen freely.

1.6.2 Redundancy

We now define functional, intrinsic and kinematic redundancy from the above-defined spaces.

Definition 1.7. A **manipulator** is denoted *intrinsically redundant* if $m < n$.

We thus note that intrinsic redundancy is a property of the manipulator design and not of the task. This allows us to design manipulators that always have a freedom that we can use to improve the performance and increase dexterity. Intrinsically redundant robots are, however, more expensive to manufacture, the complexity increases with the number of joints and so does the possibility of joint failure and other faults.

Definition 1.8. A **task** is denoted *functionally redundant* if $r < m$.

Functional redundancy is thus determined by the difference in the dimension of the operational space and the task space and does not depend on the manipulator design. If this property is present there are two main approaches that we can take; to improve design or to improve performance. One solution is to design a manipulator with fewer degrees of mobility and thus decrease production cost, complexity, and so on. On the other hand, if the manipulator has $n = m$ degrees of mobility we have one or more degrees of freedom of the end effector that can be chosen freely. This can be utilised in the motion planning and control to improve performance.

Finally a manipulator and specific task is denoted *kinematically redundant* if it is either intrinsically redundant or functionally redundant, or both. We stress, however, that in most texts on robotics, kinematic redundancy is defined in the same way as we have defined intrinsic redundancy above and in some of the papers in Part III we have adopted this notation.

1.6.3 Extended Definition of Functional Redundancy

In this section we present the basic idea of extending the definition of functional redundancy to also include a freedom in the specification of the end effector that cannot be described by the conventional framework. The details are found in the papers included in Part III. The freedom described in this section can be described as a continuous set of orientations of the end effector. We are only concerned with the end-effector orientation, but the same ideas can be applied to restrict the end-effector position.

We start by defining a continuous set of frames that satisfy some constraint on the orientation or direction of the end effector. This set is defined by restricting the allowed rotation around one or more of the coordinate axes. By restricting the rotation around different axes we can construct sets with different shapes and sizes suitable for different tasks. The size of the allowed set is then given by the size of the rotation allowed around each axis and the shape is given by the sequence of coordinate axes we choose to rotate about.

Three different sets are shown in Fig. 1.10. Note that all the sets are convex. This is an important property when we include this cone as a constraint in an optimisation problem. The different shapes shown in Fig. 1.10 are defined by different norms in $\mathbb{R}^3$. All these sets are well suited to describe the freedom that arises when a small orientation error is allowed, i.e. when a small deviation from the optimal orientation is not of concern for the task at hand.

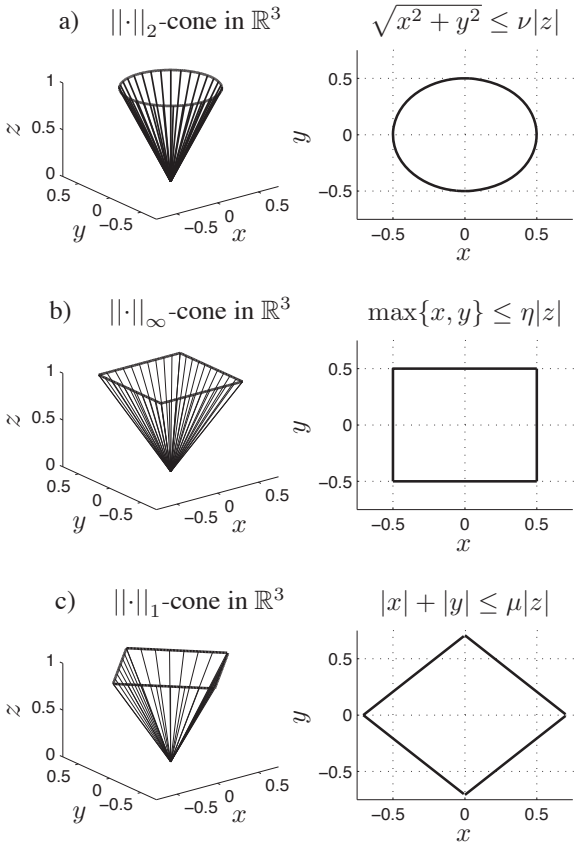


Figure 1.10: Different convex cones in $\mathbb{R}^3$.

1.6.4 Applications

The extended definition of functional redundancy can be used to improve performance for a wide variety of applications. The most apparent examples are tasks where a small error in the end-effector orientation does not affect the quality of the job performed. In this thesis we illustrate the benefits of extending the definition of functional redundancy by a spray paint example. It can be shown that allowing a small orientation error in spray paint applications does not affect the quality of the paint job. Platforms in harsh environments, such as high sea, are subject to extreme corrosion which may require the platforms to be painted with frequent intervals. Due to the large surfaces to be painted this is time consuming task and reducing the cycle time without compromising the paint quality may reduce the cost involved with painting, and also free the robots to perform other tasks.

The search for new oil fields in more remote areas will force the oil companies to operate in extremely cold areas, such as the Shtokman field explored by Statoil where temperatures reach $-45°$C. This poses enormous challenges when it comes to preventing severe damage to the equipment and an important task for the robots will therefore be to operate heating devices. A heating device mounted on a robotic manipulator is not affected by the orientation with respect to the surface to a very large extent. We can thus exploit this in the same way as for spray painting.

High pressure water blasting is another applications where the orientation is not critical and we can substantially reduce the cycle time by applying the extended definition of functional redundancy. This is also an important application, especially at sea where high pressure water blasting is used to avoid corrosion and to prepare the surfaces for painting. Other applications include arc welding, holding a camera or other measuring equipment, etc.

This kind of freedom can also be used to improve accessibility. When searching for the optimal path of the end effector in the presence of obstacles the proposed definition of functional redundancy allows us to include this freedom in the algorithms. For autonomous operating robots we need a rigorous mathematical representation in order for the path planner to search all possible paths, including the paths that exploit this freedom. In the presence of obstacles the proposed approach may thus increase the workspace of the manipulator for a specific task. The proposed formulation is also well suited for autonomous path generation as the constraints are explicit and well suited for implementation in path planners and optimisation algorithms in general.

1.6.5 Relevant Literature and Advantages of the Proposed Approach

The approach presented in this paper is based on the observation that a certain group of non-linear constraints can be transformed into a test of positive definiteness of a linearly constrained matrix. In Buss et al. (1996) the problem of determining suitable grasping forces of a robotic hand in order to balance the external forces is solved. The force of each finger is found subject to a non-linear constraint called the friction cone. The main contribution of this paper is to transform the problem of friction force limit constraints into a problem of testing for positive definiteness of a certain matrix. The problem of force optimisation is then solved as an optimisation problem on the smooth manifold of linearly constrained positive definite matrices. This guarantees globally convergent solutions that can be solved very efficiently and in real time.

Han et al. (2000) further cast the problem of friction force constraints into Linear Matrix Inequalities (LMIs) and efficient solutions are found by solving a convex optimisation problem involving LMIs. Convex optimisation problems have been extensively studied in literature and a wide range of tools can be used to find efficient algorithms and solving problems such as the existence problem and finding an initial point that satisfies the constraints. Boyd and Wegbreit (2007) present robust and computationally efficient solutions to these problems.

In Chapter 9 (From and Gravdahl, 2008d) these ideas are used to convert the problem of orientation error constraints into a test of positive definiteness of a matrix. This is possible due to the results presented in Chapter 8 (From and Gravdahl, 2007b) where it is shown how to represent a continuous set of frames representing the end-effector orientation as a

constraint directly on the four entries of the quaternion. For different types of orientation errors, a suitable matrix is found and it is shown that positive definiteness of this matrix is equivalent to an orientation that satisfies the given restrictions.

By transforming the non-linear orientation constraints into positive definiteness constraints imposed on certain matrices the problem of finding the optimal orientation is thus transformed into an optimisation problem on the smooth manifold of linearly constrained positive definite matrices. For the special case of positive definite symmetric matrices, the constraints can be written in the form of linear matrix inequalities (LMIs). The constraints can also be written as barrier functions which allows for simple and efficient implementation.

There is a huge advantage of being able to formulate the constraints as LMIs. With the recent advances in computational power, problems formulated in this form can now be solved in real time and in many cases several times faster than analytical solutions that often depend on computationally heavy matrix computations. This, together with the fact that we are guaranteed to find a solution to any convex optimisation problem, makes LMIs and barrier functions suited for solving real-time problems.

Potkonjak et al. (2000) address the problem of how to reduce the cycle time to paint a surface, and the idea of introducing the paint quality as a constraint and minimise some additional cost function is presented. This opens for the possibility of allowing a small error in the orientation of the end effector in order to increase the velocity of the paint gun, reduce torques and so on. Also, in From and Gravdahl (2007a) it is shown that by allowing an orientation error in the end-effector configuration, the speed and the quality of a spray paint job is improved. However, the problem of choosing the optimal orientation error remains unsolved as the orientation error is chosen intuitively. The approach presented is thus not suitable for implementation in an optimisation algorithm. From and Gravdahl (2008f) reformulate the problem in a convex optimisation setting which allows for a robust and optimal solution to the spray paint problem. Chapter 10 (From and Gravdahl, 2010b) presents a detailed case study of how a freedom in the specifications of the end-effector orientation can be utilised to improve performance in spray paint applications.

In Chapter 11 (From and Gravdahl, 2010a) we validate the promising theoretical results and simulations presented in From and Gravdahl (2010b) through empirical studies. The simulations presented in From and Gravdahl (2010b) suggest that the torques needed to paint a surface can be reduced with as much as 50%. It is thus important to confirm the simulation results by implementing this on a real robot. The algorithms are implemented on an ABB robot and we show that both the average torque and the maximum torque values can be reduced substantially. The theory and simulations are thus validated through successful experimental results.

1.7 Part IV - Inverse Kinematics of Manipulators with no Closed Form Solution

Existing oil and gas platforms, and probably also the facilities of the future, are extremely complex constructions which cause great challenges when it comes to accessibility. A completely automated facility without direct human intervention needs robotic solutions with access to every part of the platform. To achieve this, the use of intrinsically redundant

manipulators is inevitable. These manipulators do, however, pose significant challenges—but also great potential—when it comes to solving the inverse kinematics problem. In general the inverse kinematics of intrinsically redundant manipulators cannot be solved in a closed form, and iterative solutions are applied.

There are also other types of robotic manipulators that do not have a closed form solution to the inverse kinematics problem. In some cases, the closed form solution is not known due to the complex geometry of the robot. This is for example the case for the ABB hollow wrist. The hollow wrist has the great advantage that all cables can be put inside the wrist, instead of on the outside. This is very favourable when operating in complex environments, such as an oil platform, and also when it comes to certification for use in explosive and other critical areas. We can achieve a far more robust solution if we avoid cables on the outside of the robot.

The challenge with the ABB hollow wrist is that the axes of rotation cannot be represented as constant twists, but they depend upon the position of the wrist joints. When the axes of rotation are not constant, we cannot solve the inverse kinematics by reducing the problem into several sub-problems, known as Paden-Kahan sub-problems, and find a closed form solution in this way. As there is no known closed form solution to the inverse kinematics problem for this kind of manipulators, this problem needs to be solved iteratively. In Part IV we propose a set of algorithms where we solve the problem numerically by dividing the problem into several sub-problems which allows us to solve each sub-problem very efficiently.

1.7.1 The Inverse Geometric and Inverse Kinematics Problem

In this part of the thesis we address the inverse geometric problem, often referred to as the inverse kinematics problem. We will, however, distinguish between the two, even though this is not always done in literature. The inverse kinematics problem and the inverse geometric problem solve two different problems. Which one we choose depends on whether we want to control the robot in operational space or in joint space (Khalil and Dombre, 2002). The term inverse kinematics is also often used to cover both inverse geometric and inverse kinematics in one term. When appropriate we will also adopt this definition, as in the title of Part IV.

As a general rule we can say that the inverse kinematics problem is solved when operational space control is applied, while the inverse geometric problem is applied when joint space control is applied. We will formalise this and point out the main differences in the following.

Operational Space Control

Operational space control has the advantage that the end-effector motion is specified in the Cartesian space. One popular control scheme is the Jacobian inverse control where the transformation from operational to joint space is obtained by solving the *inverse kinematics problem*. The inverse kinematics problem finds the joint velocities from the end-effector velocities by a transformation given by the inverse Jacobian. If we assume that the desired motion is specified in the operational space by x_d, we can also easily find the actual end-effector configuration x from the joint positions by the forward kinematics. Comparing

these we can include a coordinate transformation by the inverse (analytical) Jacobian in the feed-back loop which gives us the corresponding error in joint space. The control law is then given by (Sciavicco and Siciliano, 2005)

$$\tau = K\Delta\theta = KJ^{-1}(\theta)(x_d - x) \tag{1.34}$$

for some positive definite K and where θ is the vector of joint variables. There are also several other variations of this control law. The inverse Jacobian in this form is also used in the inverse dynamic control scheme (Sciavicco and Siciliano, 2005).

Operational space control has many advantages and is fast to compute. A drawback is that it strongly depends on the inverse Jacobian and that the transformation from operational to joint space is performed inside the feedback loop and thus the recomputation time of the controller strongly depends of the complexity of this transformation (Perdereau et al., 2002). This may lead to degrading performance of the controller. Operational space controllers are however important in force control and when the end effector is in contact with the environment in general.

When kinematic redundancy is present, the inverse Jacobian approach also allows us to add a secondary criteria, such as minimising the cost function of joint velocities $\psi = \frac{1}{2}\dot{\theta}^{\mathsf{T}} W \dot{\theta}$ for some symmetric positive definite W. For $W = I$ the solution to this problem is given by the Moore-Penrose generalised inverse as $J^{\dagger} = J^{\mathsf{T}}(JJ^{\mathsf{T}})^{-1}$.

Joint Space Control

For joint space control, the transformation from operational space to joint space is obtained by solving the *inverse geometric problem*, i.e. to find the joint positions from the desired end-effector position/orientation. Then some joint space control scheme can be designed. The disadvantage of this approach is that the inverse geometric is a time-consuming problem to solve. The advantage is that the transformation from operational to joint space is moved outside the control loop. For joint space control there are a wide variety of approaches available, including standard feedback control and computed torque feedforward control, as well as robust and adaptive control schemes.

When kinematic redundancy is present, the inverse geometric approach also allows for optimising a general secondary criteria, and does not depend on finding a suitable inverse of the Jacobian, such as the Moore-Penrose generalised inverse, as for the inverse kinematics problem. Another advantage of the inverse geometric approach is that each joint can be controlled more directly and given the desired characteristics such as joint stiffness, energy consumption, maximum velocity, and obstacle avoidance. For the inverse Jacobian approach these characteristics must be added through the choice of the cost W. In some cases, such as the minimisation of energy through the Moore-Penrose inverse, this is both efficient and elegant, but for other characteristics such a Jacobian may be very hard or impossible to find.

1.7.2 Closed-form Solutions to the Inverse Geometric Problem

Most industrial manipulators with six revolute joints have a known analytical solution to the inverse geometric problem. When intersecting axes are present, this can be used

to reduce the problem to several simpler sub-problems, known as the Paden-Kahan sub-problems. The Paden-Kahan approach uses the product of exponentials formula found in Brockett (1984) and was presented in Paden and Sastry (1988). The work was based on the unpublished work of Kahan (Kahan, 1983).

The basic concept is to apply the kinematic equations to certain points on the intersecting axes and then use this to eliminate the dependence of these joints to the position of the chosen points. To be able to reduce the problem into Paden-Kahan sub-problems, some requirement must be met:

- There must be a certain number of intersecting axes.
- Each joint must be represented by the exponential map. This is the same as requiring that all links revolute around fixed axes.

A wide range of industrial manipulators have a geometry for which we can find intersecting axes. We then apply the kinematic equations to points that lie on these intersecting axes. This simplifies the equations as $\exp(\hat{\xi}\theta)p = p$ for a point p on the axis ξ eliminates the dependence of the joint angle θ. For example, we cannot solve the inverse geometric problem

$$e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}e^{\hat{\xi}_3\theta_3} = g_{03} \tag{1.35}$$

directly, but if we apply the left and the right side to a point p on ξ_3, we get

$$e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}p = g_{03}p \tag{1.36}$$

for which we have an analytical solution (Murray et al., 1994). By eliminating the dependence of certain joint variables in this way we reduce the problem to several sub-problems that can be solved analytically.

We stress that there is only a certain class of robotic manipulators for which we can apply this approach, but this class is rather big and includes some commonly used industrial manipulators. There are also other ways to derive analytical expressions of the inverse geometric problem, but most of them are based on geometric analysis and often do only apply to one specific manipulator. There is thus a need for algorithms that work also on manipulators without certain geometric properties. This is discussed in the next section.

1.7.3 Iterative Methods in Literature

There is a wide variety of methods available for solving the inverse geometric problem numerically. Most of these stem from the robot research communities, but also the computer graphics community has contributed to this field. Some of the most common approaches are Jacobian pseudoinverse methods (Whitney, 1969), Jacobian transpose methods (Balestrino et al., 1984), damped least squares methods (Nakamura and Hanafusa, 1986), conjugate gradient methods (Wang and Chen, 1991; Zhao and Badler, 1994), cyclic coordinate descent methods (Wang and Chen, 1991), and artificial intelligence methods (D'Souza et al., 2001; Oyama et al., 2001; Tevatia and Schaal, 2000). These methods can be divided into two main groups: The first group consists of Jacobian-based methods while the second is purely optimisation based.

Firstly, Jacobian based methods take advantage of the geometry of the mechanism and calculates the incremental change in the joint positions from the end-effector error. There are many variations of the Jacobian based approach but in its simplest form it is given iteratively by solving the following algorithm:

- compute the end-effector error $\Delta x = x_d - x$,
- compute the corresponding joint position error $\Delta\theta = J^{-1}\Delta x$,
- update the joint positions $\theta^{k+1} = \theta^k + \Delta\theta$.

Here x_d is the desired end-effector configuration, x the current end-effector configuration, J is the analytical Jacobian and θ is the joint positions. This will eventually converge to the correct solution. Jacobian based methods suffer from singularities due to the inversion of the Jacobian. The convergence is also poor when the initial guess is far from the solution and this may lead to oscillations.

Secondly, purely optimisation based methods do not require the Jacobian and do thus not suffer from singularities. These methods can again be divided in two main groups, the ones that require the gradient, such as Cauchy steepest descent, and the ones that do not, such as coordinate descent methods .

The steepest descent method is a popular method for the minimisation of a function of several variables (Luenberger, 2003). It is also known as the gradient method as it uses the gradient ∇f to find the search direction. The steepest descent is given by iteratively calculating the next point by

$$\theta^{k+1} = \theta^k - \alpha^k \nabla f(\theta^k)^\mathsf{T} \tag{1.37}$$

where α^k is a non-negative scalar minimising $f(\theta^k - \alpha^k \nabla f(\theta^k))$. α^k can be found in many ways. A computationally quick solution is to let α^k be a small constant. This allows us to do several iterations in a short time, at the cost of not finding the minimum value in the search direction. We can also search in the direction of the negative gradient for a minimum of this line. If this can be done in a computationally efficient way, this will give us a fast and reliable solution. Convergence to a point where $\nabla f(\theta) = 0$ can be proven (Luenberger, 2003).

Methods that do not require the gradient will often divide the problem into sub-problems that can be solved very efficiently. One such approach that we will focus on is the coordinate descent algorithm that optimises a cost function $f(\theta)$, $\theta \in \mathbb{R}^n$, by sequentially minimising with respect to each of the components, θ_i, for $i = 1 \ldots n$ (Grudic and Lawrence, 1993; Johnson, 1995; Wang and Chen, 1991; Luenberger, 2003). The convergence of coordinate descent is in general poorer than the steepest descent as only one variable is updated at each time step. Coordinate descent is, however, easy to implement and a fast solution to the sub-problem makes these algorithms relatively fast. We note that we do not need the gradient which saves a considerable amount of computational power compared to the steepest descent. The Cyclic Coordinate Descent (CCD) method performs this optimisation cyclically and is treated in detail in Welman (1993) and Wang and Chen (1991). The standard description of the CCD assumes an Euler angle representation. Johnson (1995) extend these works and formulate the CCD using the unit quaternion approach.

1.7.4 The Proposed Approach

We treat the inverse geometric problem as a pure optimisation problem and solve the problem using the cyclic coordinate descent (CCD) approach. For each joint we find the optimal joint position, assuming all other joints as fixed. By optimal we may refer to the joint position that minimises the position error of the end effector, the orientation error of the end effector, or both. Once this is found we move on to the next variable. Inspired by the work of Welman (1993) and Johnson (1995) we divide the problem into smaller subproblems that are solved iteratively. The main contribution of our approach is that, using the quaternion representation, we find that we can formulate the optimisation problem so that an analytical solution can be found. This was also done in Johnson (1995) for the position of the end effector, but we formulate such an optimisation problem also for the end-effector orientation.

The novelty of the method presented is thus that the minimum of the cost function with respect to each joint is found analytically and this is exploited to develop a set of computationally efficient algorithms. It is well known that the convergence of coordinate descent is slower than steepest descent and Newton's method. The advantage is that the analytic solution presented is a lot faster to solve than search algorithms in general. Thus, by formulation the optimisation problem so that the solution can be found in closed form for each sub-problem we develop computationally efficient algorithms, even though the convergence of each step is not as good at gradient or Jacobian based approaches.

A total of six algorithms are presented. The first three use *coordinate descent* which looks at one joint at the time. The difference between the approaches lies in the interpretation of the cost function, i.e. how the orientation and position errors are handled by the algorithm. For example, we may choose to optimise with respect to both position and orientation for each joint, or we may choose to first optimise with respect to orientation for all the joints, and then with respect to position for all the joints. In addition we investigate in detail the effect of changing the sequence of the joints to be optimised, i.e. if we start with the joint closest to the base or the end effector.

The last three methods can be looked upon as approximations of steepest descent where the gradient is estimated. The idea here is to use the optimal solution found from each joint, like for CCD, but without updating the joint positions for each joint. We then use the information about how far each joint should move to reach the optimal position to estimate the gradient. This provides us with a very efficient way to estimate the gradient as the solutions to all the sub-problems are found analytically. This gradient can then be used in for example the steepest descent approach. Thus, with this method the gradient can be found very efficiently without the need to differentiate, at the cost of finding only an approximation. It is also argued that the step size can be set as a constant. Hence, a closed form and subsequently a computationally very efficient alternative to both the search direction and the step size of the steepest descent approach is presented.

It is shown that the algorithms that approximate the steepest descent have very good convergence and reliability for difficult problems. However, for easy problems, when the initial guess is close to the solution, the convergence is better for conventional Jacobian-based algorithms than the algorithms proposed here. This corresponds with most of the literature which states that Jacobian based algorithms perform well when the initialisation point is close to the solution, but performs poorly for a bad choice of initialisation point.

For problems where Jacobian based algorithms have poor convergence or reliability, the algorithms presented here are thus a better choice. A combination of the algorithms presented and a Jacobian based method should give good and reliable performance for difficult problems but also reasonably good convergence close to the solution. We thus propose to use one of the algorithms presented in Part IV to find a configuration in the neighbourhood of the solution and then a Jacobian based approach to obtain the exact solution.

Part I

Singularity-Free Vehicle-Manipulator Modelling

Chapter 2

Singularity-Free Dynamic Equations of Vehicle-Manipulator Systems

2.1 Abstract

In this paper we derive the singularity-free dynamic equations of vehicle- manipulator systems using a minimal representation. These systems are normally modeled using Euler angles, which leads to singularities, or Euler parameters, which is not a minimal representation and thus not suited for Lagrange's equations. We circumvent these issues by introducing quasi-coordinates which allows us to derive the dynamics using minimal and globally valid non-Euclidean configuration coordinates. This is a great advantage as the configuration space of the vehicle in general is non-Euclidean. We thus obtain a computationally efficient and singularity-free formulation of the dynamic equations with the same complexity as the conventional Lagrangian approach. The closed form formulation makes the proposed approach well suited for system analysis and model-based control. This paper focuses on the dynamic properties of vehicle-manipulator systems and we present the explicit matrices needed for implementation together with several mathematical relations that can be used to speed up the algorithms. We also show how to calculate the inertia and Coriolis matrices and present these for several different vehicle-manipulator systems in such a way that this can be implemented for simulation and control purposes without extensive knowledge of the mathematical background. By presenting the explicit equations needed for implementation, the approach presented becomes more accessible and should reach a wider audience, including engineers and programmers.

Keywords: Robot modeling, vehicle-manipulator dynamics, singularities, quasi-coordinates.

2.2 Introduction

A good understanding of the dynamics of a robotic manipulator mounted on a moving vehicle is important in a wide range of applications. Especially, the use of robots in harsh and remote areas has increased the need for vehicle-robot solutions. A robotic manipulator mounted on a moving vehicle is a flexible and versatile solution well suited for these applications and will play an important role in the operation and surveillance of remotely located plants in the very near future. Recreating realistic models of for example space or deep-sea conditions is thus important. Both for simulation and for model-based control the explicit dynamic equations of vehicle-manipulator systems need to be implemented in a robust and computationally efficient way to guarantee safe testing and operation of these systems.

One example of such a system is spacecraft-manipulator systems (Egeland and Sagli, 1993; Dubowsky and Papadopoulos, 1993; Hughes, 2002; Moosavian and Papadopoulos, 2004, 2007) which are emerging as an alternative to human operation in space. Operations include assembling, repair, refuelling, maintenance, and operations of satellites and space stations. Due to the enormous risks and costs involved with launching humans into space, robotic solutions evolve as the most cost-efficient and reliable solution. However, space manipulation involves quite a few challenges. In this paper we focus on modeling spacecraft-manipulator systems, which is quite different from standard robot modeling. Firstly, the manipulator is mounted on a free-floating (unactuated) or free-flying (actuated) spacecraft. There is thus no obvious way to choose the inertial frame. Secondly, the motion of the manipulator affects the motion of the base, which results in a set of dynamic equations different from the fixed-base case due to the dynamic coupling. Finally, the free fall environment complicates the control and enhances the non-linearities in the Coriolis matrix. The framework presented in this paper is especially suited for modeling such systems, especially when applying the so-called dynamically equivalent manipulator approach (Liang et al., 1998; Parlaktuna and Ozkan, 2004). A set of minimal, singularity free dynamic equations for spacecraft-manipulator systems are presented for the first time using the proposed framework.

A second example studied in detail in this paper is the use of autonomous underwater vehicles (AUVs) with robotic arms, or underwater robotic vehicles (URVs). This is an efficient way to perform challenging tasks over a large sub-sea area. Operations at deeper water and more remote areas where humans cannot or do not want to operate, require more advanced and robust underwater systems and thus the need for continuously operating robots for surveillance, maintenance, and operation emerges (Love et al., 2004; Kitarovic et al., 2005; Antonelli, 2006; McMillan et al., 1995). We derive the minimal, singularity free dynamic equations of AUV-manipulator systems using the proposed framework, which is presented for the first time in this paper. We also show how to add the hydrodynamic effects such as added mass and damping forces.

The use of robotic manipulators on ships is another important application (Kitarovic et al., 2005; Oh et al., 2005). In From et al. (2009a) the dynamic equations were derived and the effects of the moving ship on the manipulator was analyzed. In the Ampelmann project (Salzmann, 2007) a Stewart platform is mounted on a ship and is used to compensate for the motion of the ship by keeping the platform still with respect to the world frame. This can be modeled as a 2-joint mechanism where one joint represents the un-

controllable ship motion and one joint the Stewart platform. There are also other relevant research areas where a robotic manipulator is mounted on a floating base. Lebans et al. (1997) give a cursory description of a telerobotic shipboard handling system, and Kosuge et al. (1992); Kajita and Kosuge (1997) address the control of robots floating on the water utilizing vehicle restoring forces. Another interesting research area is macro/micro manipulators (Yoshikawa et al., 1996; Bowling and Khatib, 1997) where the two manipulators in general have different dynamic properties.

It is a well known fact that the kinematics of a rigid body contains singularities if the Euler angles are used to represent the orientation of the body and the joint topology is not taken into account. One solution to this problem is to use a non-minimal representation such as the unit quaternion to represent the orientation. This will, however, increase the complexity of the implementation and because the unit quaternion is a covering manifold for the set of rotation matrices they are also subject to the unfortunate unwinding phenomenon (Bhat and Bernstein, 2000). Also, as the number of variables is not minimal, this representation cannot be used in Lagrange's equations. This is a major drawback when it comes to modeling vehicle-manipulator systems as most methods used for robot modeling are based on the Lagrangian approach. It is thus a great advantage if also the vehicle dynamics can be derived from the Lagrange equations.

The use of Lie groups and algebras as a mathematical basis for the derivation of the dynamics of multibody systems can be used to overcome this problem (Selig, 2000; Park et al., 1995). We then choose the coordinates generated by the Lie algebra as local Euclidean coordinates which allows us to describe the dynamics locally. For this approach to be valid globally the total configuration space needs to be covered by an atlas of local exponential coordinate patches. The appropriate equations must then be chosen for the current configuration. The geometric approach presented in Bullo and Lewis (2004) can then be used to obtain a globally valid set of dynamic equations on a single Lie group, such as an AUV or spacecraft with no robotic manipulator attached.

Even though combinations of Lie groups can be used to represent multibody systems, the formulation is very complex and not suited for implementation in a simulation environment. In Kwatny and Blankenship (2000) quasi-coordinates and the Lie bracket were used to derive the dynamic equations of fixed-base robotic manipulators without singularities using Poincaré's formulation of the Lagrange equations. In Kozlowski and Herman (2008); Herman and Kozlowski (2006) several control laws using a quasi-coordinate approach were presented, but only robots with conventional 1-DoF joints were considered. Common for all these methods is, however, that the configuration space of the vehicle and robot is described as $q \in \mathbb{R}^n$. This is not a problem when dealing with 1-DoF revolute or prismatic joints but more complicated joints such as ball-joints or free-floating joints then need to be modeled as compound kinematic joints (Kwatny and Blankenship, 2000), i.e., a combination of 1-DoF simple kinematic joints. For joints that use the Euler angles to represent the generalized coordinates this solution leads to singularities in the representation.

In this paper we follow the generalized Lagrangian approach presented in Duindam and Stramigioli (2007, 2008) which allows us to combine the Euclidean joints and more general joints, i.e., joints that can be described by the Lie group $SE(3)$ or one of its ten subgroups, and we extend these ideas to vehicle-manipulator systems. There are several advantages in following this approach. The use of quasi-coordinates, i.e., velocity coor-

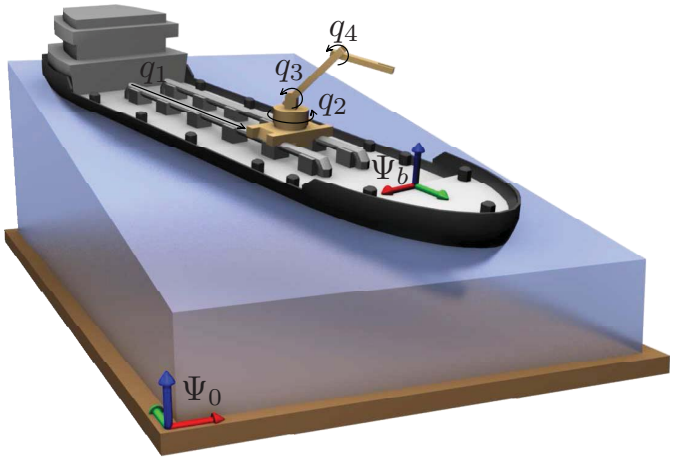


Figure 2.1: Model setup for a four-link robot attached to a vehicle, in this case a ship, with coordinate frame Ψ_b. Frame Ψ_0 denotes the inertial reference frame.

dinates that are not simply the time derivative of the position coordinates, allows us to include joints (or transformations) with a different topology than that of $\mathbb{R}^n$. For example, for an AUV we can represent the transformation from the inertial frame to the AUV body frame as a free-floating joint with configuration space $SE(3)$ and we avoid the singularity-prone kinematic relations between the inertial frame and the body frame velocities that normally arise in deriving the AUV dynamics (Fossen, 2002). This relation is subject to the well known Euler angle singularities and the dynamics are not valid globally. With our approach we thus get improved numerical stability due to the absence of singularities and, as the dynamics are valid globally, we avoid switching between different dynamic models in the implementation. This approach differs from previous work in that it allows us to derive the dynamic equations of vehicle-manipulator systems for vehicles with a configuration space different from $\mathbb{R}^n$ and thus maintains the underlying topology of the configuration space. The dynamics are expressed (locally) in exponential coordinates ϕ, but the final equations are evaluated at $\phi = 0$. This has two main advantages. Firstly, the dynamics do not depend on the local coordinates as these are eliminated from the equations and the global position and velocity coordinates are the only state variables. This makes the equations valid globally. Secondly, evaluating the equations at $\phi = 0$ greatly simplifies the dynamics and make the equations suited for implementation in simulation software. We also note that the approach is well suited for model-based control as the equations are explicit and without constraints. The fact that the configuration space of the vehicle in general is a Lie group also simplifies the implementation. Even though the expressions in the derivation of the dynamics are somewhat complex, we have several tools from the Lie theory that allows us to write the final expressions in a very simple form. We present several examples of how we can use this to simplify the dynamic equations and speed up the implementation.

The main purpose of this paper is to study systems that consist of a moving vehicle with a robotic manipulator attached to it. To the authors' best knowledge these systems

have not been studied in detail in literature using the framework presented here. There is an apparent need to be able to derive the dynamics of such systems globally and using a minimal representation, especially when it comes to formulating model-based control laws. In this paper we first present the framework, based on the approach in Duindam and Stramigioli (2007, 2008), and then show how to expand this to vehicle-manipulator systems. The use of quasi-coordinates to derive the dynamics in this way has mainly been applied to standard robotic manipulators with the extension to more general types of joints in Kwatny and Blankenship (2000); Duindam and Stramigioli (2008). However, the treatment of vehicle-manipulator systems deserves a special treatment. There are several reasons for this. Firstly, the vehicle and the manipulator may possess completely different dynamic properties. One apparent example is when the vehicle possesses a forced uncontrollable motion while the manipulator is controllable. This is the case for manipulators mounted on ships, as treated in From et al. (2009a), where the high-frequency motion of the ship is a forced motion due to the waves and wind. Spacecraft-manipulator systems are another example where the spacecraft may be unactuated and its position is determined by the robot motion. Secondly, the formulation allows us to study how the two systems, i.e., the vehicle and the manipulator, affect each other. The interaction of the two systems will depend on the inertial properties of the two systems, external forces acting on one or both systems and the type of the vehicle (floating, submerged, rolling, fixed, etc.).

The paper is organized as follows. Section 2.3 gives the detailed mathematical background for the proposed approach. This section can be skipped and practitioners mainly interested in implementation can go straight to Section 2.4 or 2.5. Section 2.4 gives the explicit dynamic equations for the AUV-manipulator dynamics along with some comments on implementing these in a simulation environment. This includes hydrodynamic and damping forces, the added mass and Coriolis matrices and other considerations that are not encountered in robot dynamics. Section 2.5 presents the dynamic equations for spacecraft-manipulator systems and the effects of a free-floating base in a free fall environment are treated in detail. The matrix representation of the dynamics and how to implement this is presented in great detail for several vehicles with different configuration spaces. This allows the readers to first analyse the dynamics of the system from the given equations and then implement this in a simulation or control environment without having to perform all the detailed computations themselves.

2.3 Dynamic Equations of Vehicle-Manipulator Systems

We extend the classical dynamic equations for a serial manipulator arm with 1-DoF joints to include the motion of the vehicle on which the manipulator is mounted. We assume that the motion of the vehicle can be described by a Lie group, i.e., $SE(3)$ or one of its ten subgroups. The most important examples of "vehicles" that have a Lie group topology are shown in Table 2.1.

2.3.1 Vehicle-Manipulator Kinematics

Consider the setup of Fig. 2.1 describing a general n-link robot manipulator arm attached to a vehicle. Choose an inertial coordinate frame Ψ_0, a frame Ψ_b rigidly attached to the

$SE(3)$	-	AUV, 6-DoF ship, aeroplane, spacecraft
$X(z)$	-	The Schönflies group
$T(3)$	-	3-DoF gantry crane
$SO(3)$	-	Spacecraft (DEM approach), ball joint
$SE(2)$	-	Ground vehicle, 3-DoF ship
$T(2)$	-	2-DoF gantry crane

Table 2.1: Lie subgroups of $SE(3)$ and corresponding "vehicles". Even though some of these can be modeled as a combination of 1-DoF Euclidean joints we consider these as vehicles and group them correspondingly. The Schönflies Group $X(z)$ represent 3-DoF translation and a 1-DoF rotation about the z-axis.

vehicle, and n frames Ψ_i (not shown) attached to each link i at the center of mass with axes aligned with the principal directions of inertia. Finally, choose a vector $q \in \mathbb{R}^n$ that describes the configuration of the n joints. Using standard notation (Murray et al., 1994), we can describe the pose of each frame Ψ_i relative to Ψ_0 as a homogeneous transformation matrix $g_{0i} \in SE(3)$ of the form

$$g_{0i} = \begin{bmatrix} R_{0i} & p_{0i} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{2.1}$$

with rotation matrix $R_{0i} \in SO(3)$ and translation vector $p_{0i} \in \mathbb{R}^3$. This pose can also be described using the vector of joint coordinates q as

$$g_{0i} = g_{0b}g_{bi} = g_{0b}g_{bi}(q). \tag{2.2}$$

The vehicle pose g_{0b} and the joint positions q thus fully determine the configuration state of the robot. Even though we use g_{0b} (6 DoF) to represent the vehicle configuration, the actual configuration space of the vehicle may be a subspace of $SE(3)$ of dimension $m < 6$. For ground vehicles the configuration space is $SE(2)$, with dimension $m = 3$, and the attitude of a spacecraft has configuration space $SO(3)$, also with dimension $m = 3$.

In a similar way, the spatial velocity of each link can be expressed using twists (Murray et al., 1994):

$$V_{0i}^0 = \begin{bmatrix} v_{0i}^0 \\ \omega_{0i}^0 \end{bmatrix} = V_{0b}^0 + V_{bi}^0 = \mathrm{Ad}_{g_{0b}}\left(V_{0b}^b + J_i(q)\dot{q}\right) \tag{2.3}$$

where v_{0i}^0 and ω_{0i}^0 are the linear and angular velocities, respectively, of link i relative to the inertial frame, $J_i(q) \in \mathbb{R}^{6\times n}$ is the geometric Jacobian of link i relative to Ψ_b, the adjoint is defined as $\mathrm{Ad}_g := \left[\begin{smallmatrix} R & \hat{p}R \\ 0 & R \end{smallmatrix}\right] \in \mathbb{R}^{6\times 6}$, and $\hat{p} \in \mathbb{R}^{3\times 3}$ is the skew-symmetric matrix such that $\hat{p}x = p \times x$ for all $p, x \in \mathbb{R}^3$. The velocity state is thus fully determined given the twist V_{0b}^b of the vehicle and the joint velocities $\dot{q}$.

In the case of $m < 6$ we define a selection matrix $H \in \mathbb{R}^{6\times m}$ such that the velocity vector of the vehicle is given by

$$V_{0b}^b = H\tilde{V}_{0b}^b, \tag{2.4}$$

where $\tilde{V}_{0b}^b \in \mathbb{R}^m$ determines the velocity state of the vehicle by selecting elements of V_{0b}^b that are not trivially zero. More generally we will write the allowed joint velocity as a vector $v_i \in \mathbb{R}^{n_i}$. The joint velocity is uniquely described by this vector and the joint twist

can be expressed in terms of this vector as $T_j^{i,i} = X_i(Q)v_i$ with $X_i(Q) \in \mathbb{R}^{6\times n_i}$ a matrix describing the instantaneously allowed twists. If X is independent of the manipulator configuration we get $H = X$. In our case we have $v_i = \dot{q}_i$ for the Euclidean joints of the manipulator and the velocity vector $v_b = \tilde{V}_{0b}^b$ for the allowed vehicle velocities. The spacial velocity when $m < 6$ is then written by

$$V_{0i}^0 = \begin{bmatrix} v_{0i}^0 \\ \omega_{0i}^0 \end{bmatrix} = V_{0b}^0 + V_{bi}^0 = \mathrm{Ad}_{g_{0b}} \left(H\tilde{V}_{0b}^b + J_i(q)\dot{q} \right). \tag{2.5}$$

2.3.2 Vehicle-Manipulator Dynamics

The previous section shows how the kinematics of the system can be naturally described in terms of the (global) state variables g_{0b}, q, V_{0b}^b, and $\dot{q}$. We now derive the dynamic equations for the system in terms of these state variables. We first assume the vehicle to be free-moving and then restrict the vehicle motion to be kinematically constrained.

To derive the dynamics of the complete mechanism (including the m-DoF between Ψ_0 and Ψ_b), we follow the generalized Lagrangian method introduced by Duindam and Stramigioli (2007, 2008). This method gives the dynamic equations for a general mechanism described by a set $Q = \{Q_i\}$ of configuration states Q_i (not necessarily Euclidean), a vector v of velocity states $v_i \in \mathbb{R}^{n_i}$, and several mappings that describe the local Euclidean structure of the configuration states and their relation to the velocity states. More precisely, the neighborhood of every state $\bar{Q}_i$ is locally described by a set of Euclidean coordinates $\phi_i \in \mathbb{R}^{n_i}$ as $Q_i = \Phi_i(\bar{Q}_i, \phi_i)$ with $\Phi_i(\bar{Q}_i, 0) = \bar{Q}_i$. $\Phi_i(\bar{Q}_i, \phi_i)$ defines a local diffeomorphism between a neighborhood of $0 \in \mathbb{R}^{n_i}$ and a neighborhood of $\bar{Q}_i$.

The trick here is to first consider Q_i a parameter, even though it strictly speaking is a state variable. We then think of the local coordinate ϕ_i as a state variable. The global coordinates v are thought of as state variables in the normal way. The Lagrangian is then written in terms of v_i for velocity and $\Phi_i(\bar{Q}_i, \phi_i)$ for position and we differentiate with respect to the velocity variable v_i and the position variable ϕ_i, not $\bar{Q}_i$ which we for now consider a parameter. Recalling that $\Phi_i(\bar{Q}_i, 0) = \bar{Q}_i$, we see that evaluating the expressions at $\phi = 0$ allows us to consider Q_i a variable and we are done. The reason we can do this is that locally the variables ϕ describe the configuration state of the system in a neighborhood of any configuration $\bar{Q}_i$.

We start by deriving an expression for the kinetic co-energy of a mechanism, expressed in coordinates Q, v, but locally parameterized by the coordinate mappings for each joint. For joints that can be described by a matrix Lie group (actually for the group of $n \times n$ nonsingular real matrices $GL(n, \mathbb{R})$), this mapping can be given by the exponential map (Murray et al., 1994). Let $A \in gl(n, \mathbb{R})$, where $gl(n, \mathbb{R})$ is the Lie Algebra of $GL(n, \mathbb{R})$. Then the exponential map $\exp(A)$ is given by

$$e^A = I + A + \frac{A^2}{2} \cdots = \sum_{n=0}^{\infty} \frac{A^n}{n!} \tag{2.6}$$

where I (no subscript) is the identity matrix. This expression is valid for all subgroups of $SE(3)$ and $SE(3)$ itself by replacing A with the matrix representation of the Lie algebra associated with the Lie group. We denote the matrix representation of the corresponding

Lie algebra by $\hat{\phi}$ and thus get

$$e^{\hat{\phi}} = I + \hat{\phi} + \frac{\hat{\phi}^2}{2} \cdots = \sum_{n=0}^{\infty} \frac{\hat{\phi}^n}{n!}. \tag{2.7}$$

Specific examples of $\hat{\phi}$ for different Lie groups are given in the following sections.

The dynamics are thus expressed in local coordinates ϕ for configuration and v for velocity, and we consider Q a parameter. After taking partial derivatives of the Lagrangian function, we evaluate the results at $\phi = 0$ (i.e., at configuration Q) to obtain the dynamics expressed in global coordinates Q and v as desired. We note that even though local coordinates ϕ appear in the derivations of the various equations, the final equations are all evaluated at $\phi = 0$ and hence these final equations do not depend on local coordinates. The global coordinates Q and v are the only dynamic state variables and the equations are valid globally, without the need for coordinate transitions between various areas of the configuration space, as is required in methods that use an atlas of local coordinate patches.

Note also that taking the partial derivatives of the Lagrangian and evaluating at $\phi = 0$ greatly simplifies (2.7) and the closed form expressions of the exponential map is not needed. We will use this observation to simplify the implementation and reduce the computational complexity of the algorithm. We will see several examples of how we can use this to simplify the expressions of the Coriolis matrices for different types of vehicles.

In general, the topology of a Lie group is not Euclidean. When deriving the dynamic equations for vehicles such as ships (Fossen, 2002), AUVs (Antonelli, 2006), and spacecraft (Hughes, 2002), this is normally dealt with by introducing a transformation matrix that relates the local and global velocity variables. However, forcing the dynamics into a vector representation in this way, without taking the topology of the configuration space into account, leads to singularities in the representation or other deficiencies. To preserve the topology of the configuration space we will use quasi-coordinates, i.e., velocity coordinates that are not simply the time-derivative of position coordinates, but given by a linear relation. Thus, there exist differentiable matrices S_i such that we can write $v_i = S_i(Q_i, \phi_i)\dot{\phi}_i$ for every Q_i. For Euclidean joints this relation is given simply by the identity map while for joints with a Lie group topology we can use the exponential map to derive this relation.

Given a mechanism with coordinates formulated in this generalized form, we can write its kinetic energy as $\mathcal{T}(Q, v) = \frac{1}{2} v^{\mathsf{T}} M(Q) v$ with $M(Q)$ the inertia matrix in coordinates Q and v the stacked velocities of the vehicle, represented by v_b, and the robot joints, represented by v_i, $i = 1 \ldots n$. The dynamics of this system then satisfy

$$M(Q)\dot{v} + C(Q, v)v = \tau \tag{2.8}$$

with τ the vector of external and control wrenches (collocated with v), and $C(Q, v)$ the matrix describing the Coriolis and centrifugal forces given by

$$\begin{aligned} C_{ij}(Q, v) := &\sum_{k,l} \left(\frac{\partial M_{ij}}{\partial \phi_k} S^{-1}_{kl} - \frac{1}{2} S^{-1}_{ki} \frac{\partial M_{jl}}{\partial \phi_k} \right) \Bigg|_{\phi=0} v_l \\ &+ \sum_{k,l,m,s} \left(S^{-1}_{mi} \left(\frac{\partial S_{mj}}{\partial \phi_s} - \frac{\partial S_{ms}}{\partial \phi_j} \right) S^{-1}_{sk} M_{kl} \right) \Bigg|_{\phi=0} v_l. \end{aligned} \tag{2.9}$$

More details and proofs can be found in references Duindam and Stramigioli (2007) and Duindam and Stramigioli (2008).

To apply this general result to systems of the form of Fig. 2.1, we write $Q = \{g_{0b}, q\}$ as the set of configuration states where g_{0b} is the Lie group $SE(3)$ or one of its sub-groups, and $v = \left[\begin{smallmatrix} \tilde{V}^b_{0b} \\ \dot{q} \end{smallmatrix}\right]$ as the vector of velocity states. The local Euclidean structure for the state g_{0b} is given by exponential coordinates (Murray et al., 1994), while the state q is itself globally Euclidean. Mathematically, we can express configurations (g_{0b}, q) around a fixed state $(\bar{g}_{0b}, \bar{q})$ as

$$g_{0b} = \bar{g}_{0b} \exp\left(\sum_{j=1}^{6} b_j (\phi_b)_j\right), \qquad q_i = \bar{q}_i + \phi_i \quad \forall\, i \in \{1, \ldots, n\}, \tag{2.10}$$

with b_j the standard basis elements of the Lie algebra $se(3)$ or one of its subalgebras. When $m < 6$ we set $b_i = 0$ for all the $n - m$ entries that are trivially zero, corresponding to Equation (2.4).

From expression (2.5) for the twist of each link in the mechanism, we can derive an expression for the total kinetic energy. Let $I_b \in \mathbb{R}^{m\times m}$ and $I_i \in \mathbb{R}^{6\times 6}$ denote the constant positive-definite diagonal inertia tensor of the base and link i (expressed in Ψ_i), respectively. The kinetic energy $\mathcal{T}_i$ of link i then follows as

$$\begin{aligned}
\mathcal{T}_i &= \frac{1}{2}\left(V^i_{0i}\right)^{\mathsf{T}} I_i V^i_{0i} \\
&= \frac{1}{2}\left(H\tilde{V}^b_{0b} + J_i(q)\dot{q}\right)^{\mathsf{T}} \mathrm{Ad}^{\mathsf{T}}_{g_{ib}} I_i \,\mathrm{Ad}_{g_{ib}} \left(H\tilde{V}^b_{0b} + J_i(q)\dot{q}\right) \\
&= \frac{1}{2}\left((\tilde{V}^b_{0b})^{\mathsf{T}} H^{\mathsf{T}} + \dot{q}^{\mathsf{T}} J_i(q)^{\mathsf{T}}\right) \mathrm{Ad}^{\mathsf{T}}_{g_{ib}} I_i \,\mathrm{Ad}_{g_{ib}} \left(H\tilde{V}^b_{0b} + J_i(q)\dot{q}\right) \\
&= \frac{1}{2}\left[\left(\tilde{V}^b_{0b}\right)^{\mathsf{T}} \quad \dot{q}^{\mathsf{T}}\right] M_i(q) \left[\begin{smallmatrix} \tilde{V}^b_{0b} \\ \dot{q} \end{smallmatrix}\right] = \frac{1}{2} v^{\mathsf{T}} M_i(q) v
\end{aligned} \tag{2.11}$$

with $M_b = \left[\begin{smallmatrix} I_b & 0 \\ 0 & 0 \end{smallmatrix}\right]$ for the vehicle and

$$M_i(q) := \begin{bmatrix} H^{\mathsf{T}} \mathrm{Ad}^{\mathsf{T}}_{g_{ib}} I_i \,\mathrm{Ad}_{g_{ib}} H & H^{\mathsf{T}} \mathrm{Ad}^{\mathsf{T}}_{g_{ib}} I_i \,\mathrm{Ad}_{g_{ib}} J_i \\ J_i^{\mathsf{T}} \mathrm{Ad}^{\mathsf{T}}_{g_{ib}} I_i \,\mathrm{Ad}_{g_{ib}} H & J_i^{\mathsf{T}} \mathrm{Ad}^{\mathsf{T}}_{g_{ib}} I_i \,\mathrm{Ad}_{g_{ib}} J_i \end{bmatrix} \in \mathbb{R}^{(m+n)\times(m+n)} \tag{2.12}$$

for the links. Here, H^{T} is the transpose of H which works fine when dealing with the Lie groups treated here, so we will use this notation throughout this paper. The total kinetic energy of the mechanism is given by the sum of the kinetic energies of the mechanism links and the vehicle, that is,

$$\mathcal{T}(q, v) = \frac{1}{2} v^{\mathsf{T}} \underbrace{\left(\begin{bmatrix} I_b & 0 \\ 0 & 0 \end{bmatrix} + \sum_{i=1}^{n} M_i(q)\right)}_{\text{inertia matrix } M(q)} v \tag{2.13}$$

with $M(q)$ the inertia matrix of the total system. Note that neither $\mathcal{T}(q, v)$ nor $M(q)$ depend on the pose g_{0b} nor the choice of inertial reference frame Ψ_0.

We can write (2.8) in block-form as follows

$$\begin{bmatrix} M_{VV} & M_{qV}^{\mathsf{T}} \\ M_{qV} & M_{qq} \end{bmatrix} \begin{bmatrix} \dot{\tilde{V}}_{0b}^{b} \\ \ddot{q} \end{bmatrix} + \begin{bmatrix} C_{VV} & C_{Vq} \\ C_{qV} & C_{qq} \end{bmatrix} \begin{bmatrix} \tilde{V}_{0b}^{b} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \tau_V \\ \tau_q \end{bmatrix} \tag{2.14}$$

with τ_V a wrench of control and external forces acting on the vehicle, expressed in coordinates Ψ_b (such that it is collocated with $\tilde{V}_{0b}^{b}$). Here the subscript V refers to the first m entries and q the remaining $n-m$ entries. To compute the matrix $C(Q,v)$ for our system, we can use the observations that $M(q)$ is independent of g_{0b}, that $S(Q,\phi)$ is independent of q, and that $S(Q,0) \equiv I$. Furthermore, the partial derivative of M with respect to ϕ_V is zero since M is independent of g_{0b}, and the second term of (2.9) is only non-zero for the C_{VV} block of $C(Q,v)$. This allows us to simplify $C(Q,v)$ slightly to

$$C_{ij}(Q,v) := \sum_{k=1}^{6+n} \left(\frac{\partial M_{ij}}{\partial \phi_k} - \frac{1}{2} \frac{\partial M_{jk}}{\partial \phi_i} \right) \Big|_{\phi=0} v_k + \sum_{k=1}^{6+n} \left(\frac{\partial S_{ij}}{\partial \phi_k} - \frac{\partial S_{ik}}{\partial \phi_j} \right) \Big|_{\phi=0} (M(q)v)_k. \tag{2.15}$$

Finally if gravitational forces are present we include these. Let the wrench associated with the gravitational force of link i with respect to coordinate frame Ψ_i be given by

$$F_g^i = \begin{bmatrix} f_g \\ \hat{r}_g^i f_g \end{bmatrix} = -m_i g \begin{bmatrix} R_{0i} e_z \\ \hat{r}_g^i R_{0i} e_z \end{bmatrix} \tag{2.16}$$

where $e_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}}$ and r_g^i is the center of mass of link i expressed in frame Ψ_i. In our case Ψ_i is chosen so that r_g^i is in the origin of Ψ_i so we have $r_g^i = 0$. The equivalent joint torque associated with link i is given by

$$\tau_g^i = J_i(q) \operatorname{Ad}_{g_{0i}}^{\mathsf{T}}(Q) F_g^i(Q) \tag{2.17}$$

where J_i is the geometric Jacobian and $\operatorname{Ad}_{g_{0i}} = \operatorname{Ad}_{g_{0b}} \operatorname{Ad}_{g_{bi}}$ is the transformation from the inertial frame to frame i. We note that both R_{0i} and $\operatorname{Ad}_{g_{0i}}$ depend on the vehicle configuration with respect to the inertial frame. The total effect of the gravity from all the links is then given by

$$n(Q) = \sum_{i=b}^{n} \tau_g^i \tag{2.18}$$

which enters (2.14) in the same way as τ.

2.3.3 Vehicles with Configuration Space $SE(3)$

The configuration space of a free-floating vehicle, such as an AUV or an aeroplane can be described by the matrix Lie group $SE(3)$. In this case we have the mapping (Duindam, 2006)

$$V_{0b}^{b} = \left(I - \frac{1}{2} \operatorname{ad}_{\phi_V} + \frac{1}{6} \operatorname{ad}_{\phi_V}^2 - \dots \right) \dot{\phi}_V \tag{2.19}$$

with $\mathrm{ad}_p = \begin{bmatrix} \hat{p}_{4\dots6} & \hat{p}_{1\dots3} \\ 0 & \hat{p}_{4\dots6} \end{bmatrix} \in \mathbb{R}^{6\times6}$ for $p \in \mathbb{R}^6$ relating the local and global velocity variables, and $\tilde{V}_{0b}^b = V_{0b}^b$. The corresponding matrices S_i can be collected in one block-diagonal matrix S given by

$$S(Q,\phi) = \begin{bmatrix} \left(I - \frac{1}{2}\,\mathrm{ad}_{\phi_V} + \frac{1}{6}\,\mathrm{ad}_{\phi_V}^2 - \dots\right) & 0 \\ 0 & I \end{bmatrix} \in \mathbb{R}^{(6+n)\times(6+n)}. \tag{2.20}$$

This shows that the choice of coordinates (Q, v) has the required form. We note that when differentiating with respect to ϕ and substituting $\phi = 0$ this simplifies the expression substantially.

The precise computational details of the partial derivatives follow the same steps as in the classical approach (Murray et al., 1994). C_{VV} depends on both the first and the second term in Equation (2.15). We have $i, j = 1 \dots 6$. Note that $\frac{\partial M_{ij}}{\partial \phi_k} = 0$ for $k < 7$ and $\frac{\partial S_{ij}}{\partial \phi_k} = 0$ for $i, j, k > 6$. This simplifies C_{VV} to

$$C_{ij}(Q,v) = \sum_{k=7}^{6+n} \left(\frac{\partial M_{ij}}{\partial \phi_k} - \underbrace{\frac{1}{2}\frac{\partial M_{jk}}{\partial \phi_i}}_{=0} \right)\Bigg|_{\phi=0} v_k + \sum_{k=1}^{6} \left(\frac{\partial S_{ij}}{\partial \phi_k} - \frac{\partial S_{ik}}{\partial \phi_j} \right)\Bigg|_{\phi=0} (M(q)v)_k. \tag{2.21}$$

Furthermore, if we write $S_b = (I - \frac{1}{2}\,\mathrm{ad}_{\phi_V} + \frac{1}{6}\,\mathrm{ad}_{\phi_V}^2 - \dots)$ we note that after differentiating and evaluating at $\phi = 0$ the matrices $\sum \frac{\partial S_{ij}}{\partial \phi_k}$ are equal to $-\frac{1}{2}\,\mathrm{ad}_{e_k}^\mathsf{T}$ where e_k is a 6-vector with 1 in the k^{th} entry and zeros elsewhere. Similarly, $\sum \frac{\partial S_{ik}}{\partial \phi_j}$ is equal to $\frac{1}{2}\,\mathrm{ad}_{e_k}^\mathsf{T}$. This is then multiplied by the k^{th} element of $M(q)v$ when differentiating with respect to ϕ_k. We then get

$$\begin{aligned} C_{VV}(Q,v) &= \sum_{k=1}^{6} \frac{\partial M_{VV}}{\partial q_k}\dot{q}_k - \frac{1}{2}\,\mathrm{ad}_{(M(q)v)_V}^\mathsf{T} - \frac{1}{2}\,\mathrm{ad}_{(M(q)v)_V}^\mathsf{T} \\ &= \sum_{k=1}^{6} \frac{\partial M_{VV}}{\partial q_k}\dot{q}_k - \mathrm{ad}_{(M(q)v)_V}^\mathsf{T} \end{aligned} \tag{2.22}$$

where $(M(q)v)_V$ is the vector of the first 6 entries (corresponding to V_{0b}^b) of the vector $M(q)v$.

$C_{Vq}(Q,v)$, i.e., $i = 1 \dots 6$ and $j = 7 \dots (6+n)$, is found in a similar manner. First we note that $\frac{\partial M_{jk}}{\partial \phi_i} = 0$ for $i = 1 \dots 6$ and that $\frac{\partial S_{ij}}{\partial \phi_k} = 0$ and $\frac{\partial S_{ik}}{\partial \phi_j} = 0$ for $j = 7 \dots (6+n)$, so only the first part is non-zero and we get

$$C_{Vq}(Q,v) = \sum_{k=1}^{6} \frac{\partial M_{Vq}}{\partial q_k}\dot{q}_k. \tag{2.23}$$

Finally, the terms C_{qV} and C_{qq} depend only on the first part of Equation (2.15) and

can be written more explicitly as (From et al., 2009a)

$$C_{qV} = \sum_{k=1}^{n} \frac{\partial M_{qV}}{\partial q_k}\dot{q}_k - \frac{1}{2}\frac{\partial^{\mathsf{T}}}{\partial q}\left(\begin{bmatrix} M_{VV} & M_{qV}^{\mathsf{T}} \end{bmatrix}\begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix}\right), \tag{2.24}$$

$$C_{qq} = \sum_{k=1}^{n} \frac{\partial M_{qq}}{\partial q_k}\dot{q}_k - \frac{1}{2}\frac{\partial^{\mathsf{T}}}{\partial q}\left(\begin{bmatrix} M_{qV} & M_{qq} \end{bmatrix}\begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix}\right). \tag{2.25}$$

The C-matrix is thus given by

$$C(Q,v) = \sum_{k=1}^{n} \frac{\partial M}{\partial q_k}\dot{q}_k - \frac{1}{2}\begin{bmatrix} 2\,\mathrm{ad}^{\mathsf{T}}_{(M(q)v)_V} & 0 \\ \frac{\partial^{\mathsf{T}}}{\partial q}\left(\begin{bmatrix} M_{VV} & M_{qV}^{\mathsf{T}} \end{bmatrix}\begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix}\right) & \frac{\partial^{\mathsf{T}}}{\partial q}\left(\begin{bmatrix} M_{qV} & M_{qq} \end{bmatrix}\begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix}\right) \end{bmatrix}. \tag{2.26}$$

2.3.4 Vehicles with Configuration Space $SO(3)$

The dynamics of a vehicle-manipulator system for a vehicle with configuration space $SO(3)$ are derived in the same way. The velocity state is thus fully determined by only three variables and we choose H so that

$$V_{0b}^b = H\tilde{V}_{0b}^b \tag{2.27}$$

with

$$H = \begin{bmatrix} 0_{3\times 3} \\ I_{3\times 3} \end{bmatrix}. \tag{2.28}$$

We then get

$$\tilde{V}_{0b}^b = \left(I - \frac{1}{2}\hat{\phi}_V + \frac{1}{6}\hat{\phi}_V^2 - \dots\right)\dot{\phi}_V. \tag{2.29}$$

The corresponding matrices S_i can be collected in one block-diagonal matrix S given by

$$S(Q,\phi) = \begin{bmatrix} \left(I - \frac{1}{2}\hat{\phi}_V + \frac{1}{6}\hat{\phi}_V^2 - \dots\right) & 0 \\ 0 & I \end{bmatrix} \in \mathbb{R}^{(3+n)\times(3+n)}. \tag{2.30}$$

We note that when differentiating with respect to ϕ and substituting $\phi = 0$, once again this simplifies the expression substantially. The precise computational details of the partial derivatives follow the same steps as for the $SE(3)$ case except for C_{VV}. Note that $\frac{\partial M_{ij}}{\partial \phi_k} = 0$ for $k < 4$ and $\frac{\partial S_{ij}}{\partial \phi_k} = 0$ for $i, j, k > 3$. When differentiating and evaluating at $\phi = 0$ the matrices $\sum \frac{\partial S_{ij}}{\partial \phi_k}$ are equal to $\frac{1}{2}\hat{e}_k$ where e_k is a 3-vector with 1 in the k^{th} entry and zeros elsewhere. Similarly, $\sum \frac{\partial S_{ik}}{\partial \phi_j}$ is equal to $-\frac{1}{2}\hat{e}_k$. We then get

$$C_{VV}(Q,v) = \sum_{k=1}^{6} \frac{\partial M_{VV}}{\partial q_k}\dot{q}_k + \widehat{(M(q)v)}_{\tilde{V}} \tag{2.31}$$

where $(M(q)v)_{\tilde{V}}$ is the vector of the first three entries of the vector $M(q)v$ (corresponding to $\tilde{V}_{0b}^b$) and $\hat{p} \in \mathbb{R}^{3\times 3}$ is the skew-symmetric matrix such that $\hat{p}x = p \times x$ for all $p, x \in \mathbb{R}^3$.

2.3.5 Summary

Table 2.2 shows the mapping from local to global velocity coordinates and the corresponding C-matrices for different Lie Groups.

Lie Group	S_{VV}	C
$SE(3)$	$I - \frac{1}{2}\operatorname{ad}_{\phi_V} + \frac{1}{6}\operatorname{ad}^2_{\phi_V} - \dots$	$\sum_{k=1}^{n} \frac{\partial M}{\partial q_k}\dot{q}_k - \frac{1}{2}\begin{bmatrix} 2\operatorname{ad}^{\mathsf{T}}_{(M(q)v)_V} & 0 \\ A & B \end{bmatrix}$
$X(z)$	$I_{4\times 4}$	$\sum_{k=1}^{n} \frac{\partial M}{\partial q_k}\dot{q}_k - \frac{1}{2}\begin{bmatrix} 0 & 0 \\ A & B \end{bmatrix}$
$T(3)$,$SE(2)$	$I_{3\times 3}$	$\sum_{k=1}^{n} \frac{\partial M}{\partial q_k}\dot{q}_k - \frac{1}{2}\begin{bmatrix} 0 & 0 \\ A & B \end{bmatrix}$
$SO(3)$	$I - \frac{1}{2}\hat{\phi}_V + \frac{1}{6}\hat{\phi}^2_V - \dots$	$\sum_{k=1}^{n} \frac{\partial M}{\partial q_k}\dot{q}_k - \frac{1}{2}\begin{bmatrix} -2\widehat{(M(q)v)}_{\tilde{V}} & 0 \\ A & B \end{bmatrix}$
$T(2)$,$C(1)$	$I_{2\times 2}$	$\sum_{k=1}^{n} \frac{\partial M}{\partial q_k}\dot{q}_k - \frac{1}{2}\begin{bmatrix} 0 & 0 \\ A & B \end{bmatrix}$
$T(1)$,H,$SO(2)$	$I_{1\times 1}$	$\sum_{k=1}^{n} \frac{\partial M}{\partial q_k}\dot{q}_k - \frac{1}{2}\begin{bmatrix} 0 & 0 \\ A & B \end{bmatrix}$
	$A = \frac{\partial^{\mathsf{T}}}{\partial q}\left(\begin{bmatrix} M_{VV} & M_{qV}^{\mathsf{T}} \end{bmatrix}\begin{bmatrix} \tilde{V}^b_{0b} \\ \dot{q} \end{bmatrix}\right)$	$B = \frac{\partial^{\mathsf{T}}}{\partial q}\left(\begin{bmatrix} M_{qV} & M_{qq} \end{bmatrix}\begin{bmatrix} \tilde{V}^b_{0b} \\ \dot{q} \end{bmatrix}\right)$

Table 2.2: The Coriolis matrix for different Lie subgroups of $SE(3)$.

2.4 AUV-Manipulator Systems

We start by presenting the state of the art dynamic equations of an AUV-manipulator system as it is normally presented in literature. It is well known that these are not valid globally due to the Euler angle singularity that arises when transforming from local to global velocity variables. Next, we show how to re-write the dynamics using the proposed framework in order to avoid the singularities. The dynamic equations have approximately the same complexity and are better suited for simulation and easier to implement. One drawback of the proposed approach is that the matrix $L = \dot{M} - 2C$ is not skew symmetric. This is a desired property in Lyapunov-based controller design but not in model-based controller design or simulation environments, for which computational speed, robustness, and ease are of higher importance.

2.4.1 State of the Art AUV Dynamics

A wide range of dynamical systems can be described by the Euler-Lagrange equations (Goldstein et al., 2001)

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}(x,\dot{x})\right) - \frac{\partial \mathcal{L}}{\partial x}(x,\dot{x}) = \tau \tag{2.32}$$

where $x \in \mathbb{R}^n$ is a vector of generalized coordinates, $\tau \in \mathbb{R}^n$ are the generalized forces and

$$\mathcal{L}(x,\dot{x}) \,:\, \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} := \mathcal{T}(x,\dot{x}) - \mathcal{V}(x). \tag{2.33}$$

Here, $\mathcal{T}(x,\dot{x})$ is the kinetic and $\mathcal{V}(x)$ the potential energy function. We assume that the kinetic energy function is positive definite and in the form

$$\mathcal{T}(x,\dot{x}) := \frac{1}{2}\dot{x}^{\mathsf{T}} M(x)\dot{x}. \tag{2.34}$$

where $M(x)$ is the inertia matrix. For a kinetic energy function on this form we can recast the Euler-Lagrange equations (2.32) into the equivalent form

$$M_{RB}(x)\ddot{x} + C_{RB}(x,\dot{x})\dot{x} + n(x) = \tau \tag{2.35}$$

where $C_{RB}(x,\dot{x})$ is the Coriolis and centripetal matrix and $n(x)$ is the potential forces vector defined as

$$n(x) := \frac{\partial \mathcal{V}(x)}{\partial x}. \tag{2.36}$$

The Coriolis and centripetal matrix is normally obtained by the Christoffel symbols of the first kind as (Egeland and Gravdahl, 2003)

$$C_{RB}(x,\dot{x}) := \{c_{ij}\} = \left\{\sum_{k=1}^{n} \Gamma_{ijk}\dot{x}_k\right\}, \tag{2.37}$$

$$\Gamma_{ijk} := \frac{1}{2}\left(\frac{\partial m_{ij}}{\partial x_k} + \frac{\partial m_{ik}}{\partial x_j} - \frac{\partial m_{kj}}{\partial x_i}\right) \tag{2.38}$$

where $M(x) = \{m_{ij}\}$. When representing the dynamic equations using generalized coordinates we implicitly introduce non-inertial frames in which we represent the inertial properties of the rigid bodies. The Coriolis matrix arises as a result of these non-inertial frames. We note that there are several ways to define the Coriolis matrix so that $C_{ij}(x,\dot{x})\dot{x}_j = \Gamma_{ijk}\dot{x}_j\dot{x}_k$ is satisfied. Later, we will see that in deriving the dynamics using a different framework we get a different Coriolis matrix with different properties. Normally the terms where $i \neq j$ are identified with the Coriolis forces and $i = j$ with the centrifugal forces.

In addition, for floating or submerged vehicles we need to add the hydrodynamic forces and moments. The damping forces are collected in the damping matrix D and the restoring forces (weight and buoyancy) are normally included in $n(\eta)$. Furthermore, the motion of the AUV will generate a flow in the surrounding fluid. This is known as added mass. For completely submerged vehicles operating at low velocities the added mass is given

by a constant matrix $M_A = M_A^\mathsf{T} > 0$. The corresponding Coriolis matrix is given by $C_A = -C_A^\mathsf{T}$ and is found in the same way as C_{RB} by replacing M_{RB} with M_A (Fossen, 2009). We also add environmental disturbances such as currents.

The dynamics of underwater vehicles are usually given as (Fossen, 2002)

$$\dot{\eta} = J(\eta)\nu, \tag{2.39}$$

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + n(\eta) = \tau \tag{2.40}$$

where $\eta = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^\mathsf{T}$ is the position and orientation of the vessel given in the inertial frame and $\nu = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^\mathsf{T}$ is the linear and angular velocities given in the body frame. $D(\nu)\nu$ is the damping and friction matrix, $M = M_{RB} + M_A$ and $C(\nu) = C_{RB}(\nu) + C_A(\nu)$.

The velocity transformation matrix $J(\eta)$ in (2.39) transforms the velocities from the body frame to the inertial frame and is defined as

$$J(\eta) = \begin{bmatrix} R_{0b}(\Theta) & 0 \\ 0 & T_\Theta(\Theta) \end{bmatrix} \tag{2.41}$$

where $R_{0b}(\Theta)$ is the rotation matrix and depends only on the orientations of the vessel given by the Euler angles $\Theta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^\mathsf{T}$, represented in the reference frame. $T_\Theta(\Theta)$ is given by (zyx-sequence)

$$T_\Theta(\Theta) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix}. \tag{2.42}$$

We note that $T_\Theta(\Theta)$, and thus also $J(\eta)$, are not defined for $\theta = \pm\frac{\pi}{2}$. This is the well known Euler angle singularity for the zyx-sequence. The inverse mappings $T_\Theta^{-1}(\Theta)$ and $J^{-1}(\eta)$ are defined for all $\theta \in \mathbb{R}$ but singular for $\theta = \pm\frac{\pi}{2}$.

This singularity can be removed from the operational space by deriving the kinematic equations using two Euler angle representations with different singularities and switching between these two representations. It can also be avoided using the unit quaternion representation, which does not have a singularity at the cost of introducing a fourth parameter to describe the orientation. The unit quaternion representation is computationally challenging when it comes to integration and normalization. Also, in computing the Euler angles from the quaternions the Euler angle singularity is present and precautions against computational errors close to this singularity must be taken.

We note that the representation $\nu = \begin{bmatrix} x & y & z & \eta & \epsilon_1 & \epsilon_2 & \epsilon_3 \end{bmatrix}^\mathsf{T}$ where $Q = \begin{bmatrix} \eta & \epsilon_1 & \epsilon_2 & \epsilon_3 \end{bmatrix}^\mathsf{T}$ is the unit quaternion cannot be used in the Lagrangian approach since it is defined by 7 parameters. These parameters are hence not generalized coordinates.

We will assume that the ocean current ν_c is expressed in the inertial frame. Then the relative velocity in the body-fixed frame is given by

$$\nu_r = \nu - R_{0b}\nu_c. \tag{2.43}$$

The effects of the current are then included in the dynamics by using ν_r in the derivation of the Coriolis and centripetal matrices and the damping terms.

The relationship between the wrench acting on the vehicle τ and the control input of the thrusters u_V is highly non-linear. However, it is common to approximate this with a linear relation

$$\tau = Bu_V \tag{2.44}$$

where $B \in \mathbb{R}^{6\times p_u}$ is a known constant matrix, u_V is the p_u-dimensional vector of control inputs and p_u is the number of thrusters, rudders, sterns, etc.

We can rewrite the dynamics using general coordinates η, eliminating the body frame coordinates ν from the equations. We then get

$$\tilde{M}(\eta)\ddot{\eta} + \tilde{C}(\eta,\dot{\eta})\dot{\eta} + \tilde{D}(\eta,\dot{\eta})\dot{\eta} + \tilde{n}(\eta) = \tilde{\tau} \tag{2.45}$$

where

$$\tilde{M}(\eta) = J^{-\mathsf{T}}(\eta)MJ^{-1}(\eta), \tag{2.46}$$

$$\tilde{n}(\eta) = J^{-\mathsf{T}}(\eta)n(\eta), \tag{2.47}$$

$$\tilde{\tau} = J^{-\mathsf{T}}(\eta)\tau, \tag{2.48}$$

$$\tilde{D}(\eta,\dot{\eta}) = J^{-\mathsf{T}}(\eta)D(J^{-1}(\eta)\dot{\eta})J^{-1}(\eta), \tag{2.49}$$

$$\tilde{C}(\eta,\dot{\eta})\dot{\eta} = J^{-\mathsf{T}}(\eta)\left[C(J^{-1}(\eta)\eta) - MJ^{-1}(\eta)\dot{J}(\eta)\right]J^{-1}(\eta). \tag{2.50}$$

Note that the Equations (2.45-2.50) are only valid when $J^{-1}(\eta)$ is non-singular, i.e., for $\theta \neq \pm\frac{\pi}{2}$.

To formulate the Lagrange equations in a body-fixed coordinate frame directly we need to circumvent the fact the $\int_0^t \nu \mathrm{d}t$ has no physical meaning. We do this by rewriting the Langrange equations using quasi-coordinates. Write $\nu_1 = \begin{bmatrix} u & v & w \end{bmatrix}^{\mathsf{T}}$ and $\nu_2 = \begin{bmatrix} p & q & r \end{bmatrix}^{\mathsf{T}}$ and similarly for τ. Then the dynamics can be written as (Meirovich and Kwak, 1989)

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{T}}{\partial \nu_1}\right) + \hat{\nu}_2\frac{\partial \mathcal{T}}{\partial \nu_1} = \tau_1 \tag{2.51}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{T}}{\partial \nu_2}\right) + \hat{\nu}_2\frac{\partial \mathcal{T}}{\partial \nu_2} + \hat{\nu}_1\frac{\partial \mathcal{T}}{\partial \nu_1} = \tau_2. \tag{2.52}$$

We note that the dynamic equations are independent of the position vector η and the gravitational forces are thus not included in the dynamics. We thus need to augment the equations with (2.39) to get a complete description of the state space. Once again this introduces a singularity in the equations.

2.4.2 State of the Art AUV-Manipulator Dynamics

The dynamics of an AUV-manipulator system is given by (Antonelli, 2006)

$$\dot{\xi} = J(\xi)\zeta, \tag{2.53}$$

$$M(q)\dot{\zeta} + C(q,\zeta)\zeta + D(q,\zeta)\zeta + n(q,R_{0b}) = \tau \tag{2.54}$$

where $\xi = \begin{bmatrix}\eta^\mathsf{T} & q^\mathsf{T}\end{bmatrix}^\mathsf{T}$, $\zeta = \begin{bmatrix}\nu^\mathsf{T} & \dot{q}^\mathsf{T}\end{bmatrix}^\mathsf{T}$, $M(q) \in \mathbb{R}^{(6+n)\times(6+n)}$ is the inertia matrix including added mass, $C(q,\zeta) \in \mathbb{R}^{(6+n)\times(6+n)}$ is the Coriolis and centripetal matrix and $D(q,\zeta) \in \mathbb{R}^{(6+n)\times(6+n)}$ is the matrix representing the dissipative forces. τ is the vector of forces and moments working on the mechanism and is given by

$$\tau = \begin{bmatrix}\tau_V \\ \tau_q\end{bmatrix} = \begin{bmatrix}B & 0 \\ 0 & I\end{bmatrix} u \tag{2.55}$$

where $u = \begin{bmatrix}u_V^\mathsf{T} & u_q^\mathsf{T}\end{bmatrix}^\mathsf{T}$ is the control input. The velocity transformation matrix is given by

$$J(\xi) = \begin{bmatrix}R_{0b}(\Theta) & 0 & 0 \\ 0 & T_\Theta(\Theta) & 0 \\ 0 & 0 & I\end{bmatrix}. \tag{2.56}$$

2.4.3 The Proposed Approach

In this section we show how to derive the AUV-manipulator dynamics without the presence of singularities. The inertia matrix of the AUV is derived in two steps. First, M_{RB} is found from (2.13). Then the added mass $M_A = M_A^\mathsf{T} > 0$ is found from the hydrodynamic properties and we get $M = M_{RB} + M_A$. We can now use M instead of M_{RB} to derive the Coriolis and centripetal matrix (Fossen, 2002) which gives us $C = C_{RB} + C_A$. As the configuration space of an AUV can be described by the matrix Lie group $SE(3)$ we get (following the mathematics of Equations (2.19-2.25)) the Coriolis matrix

$$C(Q,v) = \sum_{k=1}^{n} \frac{\partial M}{\partial q_k}\dot{q}_k - \frac{1}{2}\begin{bmatrix} 2\,\mathrm{ad}^\mathsf{T}_{(M(q)v)_V} & 0 \\ \frac{\partial^\mathsf{T}}{\partial q}\left(\begin{bmatrix}M_{VV} & M_{qV}^\mathsf{T}\end{bmatrix}\begin{bmatrix}V_{0b}^b \\ \dot{q}\end{bmatrix}\right) & \frac{\partial^\mathsf{T}}{\partial q}\left(\begin{bmatrix}M_{qV} & M_{qq}\end{bmatrix}\begin{bmatrix}V_{0b}^b \\ \dot{q}\end{bmatrix}\right)\end{bmatrix}. \tag{2.57}$$

The dynamic equations can now be written as

$$M(Q)\dot{v} + C(Q,v)v + D(v)v + n(Q) = \tau. \tag{2.58}$$

Here, $v = \begin{bmatrix}(V_{0b}^b)^\mathsf{T} & \dot{q}^\mathsf{T}\end{bmatrix}^\mathsf{T}$ where V_{0b}^b is the velocity state of the AUV and $\dot{q}$ the velocity state of the manipulator, and $Q = \{g_{0b}, q\}$ where $g_{0b} \in SE(3)$ determines the configuration space of the AUV (non-Euclidean) and q the configuration space of the manipulator (Euclidean). We note that the singularity in (2.53) is eliminated and the state space (Q,v) is valid globally. $D(v)$ and $n(Q)$ are found in the same way as for the conventional approach. Specifically, $n(Q)$ is found by (2.18). In the following we make some brief remarks on implementing the dynamic equations in a software environment.

Computing the Partial derivatives of $M(q_1,\ldots,q_n)$

The partial derivatives of the inertia matrix with respect to $q_1,\ldots,q_n$ are computed by

$$\frac{\partial M(q_1,\ldots,q_n)}{\partial q_k} = \sum_{i=k}^{n}\left(\begin{bmatrix} I \\ J_i^{\mathsf{T}}\end{bmatrix}\left[\frac{\partial^{\mathsf{T}}\operatorname{Ad}_{g_{ib}}}{\partial q_k} I_i \operatorname{Ad}_{g_{ib}} + \operatorname{Ad}_{g_{ib}}^{\mathsf{T}} I_i \frac{\partial \operatorname{Ad}_{g_{ib}}}{\partial q_k}\right]\begin{bmatrix} I & J_i\end{bmatrix}\right) \quad (2.59)$$

$$+\sum_{i=k+1}^{n}\begin{bmatrix} 0 & \operatorname{Ad}_{g_{ib}}^{\mathsf{T}} I_i \operatorname{Ad}_{g_{ib}} \frac{\partial J_i}{\partial q_k} \\ \frac{\partial^{\mathsf{T}} J_i}{\partial q_k}\operatorname{Ad}_{g_{ib}}^{\mathsf{T}} I_i \operatorname{Ad}_{g_{ib}} & \frac{\partial^{\mathsf{T}} J_i}{\partial q_k}\operatorname{Ad}_{g_{ib}}^{\mathsf{T}} I_i \operatorname{Ad}_{g_{ib}} J_i + J_i^{\mathsf{T}} \operatorname{Ad}_{g_{ib}}^{\mathsf{T}} I_i \operatorname{Ad}_{g_{ib}} \frac{\partial J_i}{\partial q_k}\end{bmatrix}.$$

Computing the Partial derivatives of $\operatorname{Ad}_{g_{ij}}$

The main computational burden is on the computation of the partial derivatives of M with respect to q for which we need the partial derivatives of the adjoint matrices, also with respect to q. To compute these one can use a relatively simple relation. If we express the velocity of joint k as $V_{(k-1)k}^{(k-1)} = X_k \dot{q}_k$ for constant X_k, then the following holds:

Proposition 2.1. *The partial derivatives of the adjoint matrix is given by*

$$\frac{\partial \operatorname{Ad}_{g_{ij}}}{\partial q_k} = \begin{cases} \operatorname{Ad}_{g_{i(k-1)}} \operatorname{ad}_{X_k} \operatorname{Ad}_{g_{(k-1)j}} & \text{for } i < k \le j, \\ -\operatorname{Ad}_{g_{i(k-1)}} \operatorname{ad}_{X_k} \operatorname{Ad}_{g_{(k-1)j}} & \text{for } j < k \le i, \\ 0 & \text{otherwise.}\end{cases}$$

Proof. To prove this, we start by writing out the spatial velocity of frame Ψ_k with respect to $\Psi_{(k-1)}$ when $i < k \le j$:

$$\hat{X}_k \dot{q}_k = \hat{V}_{(k-1)k}^{(k-1)} = \dot{g}_{(k-1)k} g_{(k-1)k}^{-1} = \frac{\partial g_{(k-1)k}}{\partial q_k} g_{k(k-1)} \dot{q}_k$$

where $\hat{X} := \left[\begin{smallmatrix} \hat{X}_\omega & X_v \\ 0 & 0\end{smallmatrix}\right]$. If we compare the first and the last terms, we get

$$\frac{\partial R_{(k-1)k}}{\partial q_k} = \hat{X}_\omega R_{(k-1)k}, \quad (2.60)$$

$$\frac{\partial p_{(k-1)k}}{\partial q_k} = \hat{X}_\omega p_{(k-1)k} + X_v. \quad (2.61)$$

We can use this relation in the expression for the partial derivative of $\operatorname{Ad}_{g_{(k-1)k}}$:

$$\begin{aligned}\frac{\partial \operatorname{Ad}_{g_{(k-1)k}}}{\partial q} &= \begin{bmatrix} \frac{\partial R_{(k-1)k}}{\partial q_k} & \frac{\hat{p}_{(k-1)k}}{\partial q_k} R_{(k-1)k} + \hat{p}_{(k-1)k} \frac{\partial R_{(k-1)k}}{\partial q_k} \\ 0 & \frac{\partial R_{(k-1)k}}{\partial q_k}\end{bmatrix} \\ &= \begin{bmatrix} \hat{X}_\omega & \hat{X}_v \\ 0 & \hat{X}_\omega\end{bmatrix}\begin{bmatrix} R_{(k-1)k} & \hat{p}_{(k-1)k} R_{(k-1)k} \\ 0 & R_{(k-1)k}\end{bmatrix} \\ &= \operatorname{ad}_{X_k} \operatorname{Ad}_{g_{(k-1)k}}. \end{aligned} \quad (2.62)$$

It is now straight forward to show that

$$\begin{aligned}\frac{\partial \operatorname{Ad}_{g_{ij}}}{\partial q_k} &= \operatorname{Ad}_{g_{i(k-1)}} \frac{\partial \operatorname{Ad}_{g_{(k-1)k}}}{\partial q_k} \operatorname{Ad}_{g_{kj}} \\ &= \operatorname{Ad}_{g_{i(k-1)}} \operatorname{ad}_{X_k} \operatorname{Ad}_{g_{(k-1)k}} \operatorname{Ad}_{g_{kj}} \\ &= \operatorname{Ad}_{g_{i(k-1)}} \operatorname{ad}_{X_k} \operatorname{Ad}_{g_{(k-1)j}}. \end{aligned} \quad (2.63)$$

The proof is similar for $j < k \leq i$. The details are found in 2.9.1. □

Computing the Jacobian and its Partial Derivatives

The Jacobian J_i of link i is given by

$$J_i(q) = \begin{bmatrix} X_1 & \mathrm{Ad}_{g_{b1}} X_2 & \mathrm{Ad}_{g_{b2}} X_3 & \cdots & \mathrm{Ad}_{g_{b(i-1)}} X_i & 0 & \cdots & 0 \end{bmatrix}. \tag{2.64}$$

When the partial derivatives of the adjoint map are found we can also use these to find the partial derivatives of the Jacobian, i.e.,

$$\frac{\partial J_i}{\partial q_k} = \begin{bmatrix} 0_{(k+1)\times 6} & \frac{\partial \mathrm{Ad}_{g_{bk}}}{\partial q_k} X_{k+1} & \frac{\partial \mathrm{Ad}_{g_{b(k+1)}}}{\partial q_k} X_{k+2} & \cdots & \frac{\partial \mathrm{Ad}_{g_{b(i-1)}}}{\partial q_k} X_5 & 0_{(6-i)\times 6} \end{bmatrix} \tag{2.65}$$

For the special case when the twist of each joint cannot be represented as a constant vector the computation is somewhat more involved. The proposed framework does, however, allow for joints with non-constant twists. This is shown in 2.9.2.

Implementation

We first define the vector

$$(M(q)v)_V = \begin{bmatrix} (M(q)v)_1 \\ (M(q)v)_2 \\ \vdots \\ (M(q)v)_m \end{bmatrix} = \begin{bmatrix} M_{VV} & M_{qV}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix}. \tag{2.66}$$

This gives the adjoint part of the second part of (2.57) as

$$\mathrm{ad}_{(M(q)v)_V} = \tag{2.67}$$
$$\begin{bmatrix} 0 & -(M(q)v)_6 & (M(q)v)_5 & 0 & -(M(q)v)_3 & (M(q)v)_2 \\ (M(q)v)_6 & 0 & -(M(q)v)_4 & (M(q)v)_3 & 0 & -(M(q)v)_1 \\ -(M(q)v)_5 & (M(q)v)_4 & 0 & -(M(q)v)_2 & (M(q)v)_1 & 0 \\ 0 & 0 & 0 & 0 & -(M(q)v)_6 & (M(q)v)_5 \\ 0 & 0 & 0 & (M(q)v)_6 & 0 & -(M(q)v)_4 \\ 0 & 0 & 0 & -(M(q)v)_5 & (M(q)v)_4 & 0 \end{bmatrix}.$$

The lower part of the matrix in the second term in (2.57) is calculated in the following way

$$\frac{\partial^{\mathsf{T}}}{\partial q}\left(\begin{bmatrix} M_{VV} & M_{qV}^{\mathsf{T}} \end{bmatrix}\begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix}\right) = \begin{bmatrix} \frac{\partial (M(q)v)_1}{\partial q_1} & \frac{\partial (M(q)v)_2}{\partial q_1} & \cdots & \frac{\partial (M(q)v)_6}{\partial q_1} \\ \frac{\partial (M(q)v)_1}{\partial q_2} & \frac{\partial (M(q)v)_2}{\partial q_2} & \cdots & \frac{\partial (M(q)v)_6}{\partial q_2} \\ \vdots & & \ddots & \vdots \\ \frac{\partial (M(q)v)_1}{\partial q_n} & \frac{\partial (M(q)v)_2}{\partial q_n} & \cdots & \frac{\partial (M(q)v)_6}{\partial q_n} \end{bmatrix} \tag{2.68}$$
$$= \begin{bmatrix} \sum_{i=1}^{6+n} \frac{\partial M_{1i}(q)}{\partial q_1} v_i & \sum_{i=1}^{6+n} \frac{\partial M_{2i}(q)}{\partial q_1} v_i & \cdots & \sum_{i=1}^{6+n} \frac{\partial M_{6i}(q)}{\partial q_1} v_i \\ \sum_{i=1}^{6+n} \frac{\partial M_{1i}(q)}{\partial q_2} v_i & \sum_{i=1}^{6+n} \frac{\partial M_{2i}(q)}{\partial q_2} v_i & \cdots & \sum_{i=1}^{6+n} \frac{\partial M_{6i}(q)}{\partial q_2} v_i \\ \vdots & & \ddots & \vdots \\ \sum_{i=1}^{6+n} \frac{\partial M_{1i}(q)}{\partial q_n} v_i & \sum_{i=1}^{6+n} \frac{\partial M_{2i}(q)}{\partial q_n} v_i & \cdots & \sum_{i=1}^{6+n} \frac{\partial M_{6i}(q)}{\partial q_n} v_i \end{bmatrix}$$

$$\frac{\partial^{\mathsf{T}}}{\partial q}\left(\begin{bmatrix} M_{qV} & M_{qq}\end{bmatrix}\begin{bmatrix} V_{0b}^{b} \\ \dot{q}\end{bmatrix}\right) \tag{2.69}$$
$$= \begin{bmatrix} \sum_{i=1}^{6+n}\frac{\partial M_{(m+1)i}(q)}{\partial q_1}v_i & \sum_{i=1}^{6+n}\frac{\partial M_{(m+2)i}(q)}{\partial q_1}v_i & \cdots & \sum_{i=1}^{6+n}\frac{\partial M_{(m+n)i}(q)}{\partial q_1}v_i \\ \sum_{i=1}^{6+n}\frac{\partial M_{(m+1)i}(q)}{\partial q_2}v_i & \sum_{i=1}^{6+n}\frac{\partial M_{(m+2)i}(q)}{\partial q_2}v_i & \cdots & \sum_{i=1}^{6+n}\frac{\partial M_{(m+n)i}(q)}{\partial q_2}v_i \\ \vdots & & \ddots & \vdots \\ \sum_{i=1}^{6+n}\frac{\partial M_{(m+1)i}(q)}{\partial q_n}v_i & \sum_{i=1}^{6+n}\frac{\partial M_{(m+2)i}(q)}{\partial q_n}v_i & \cdots & \sum_{i=1}^{6+n}\frac{\partial M_{(m+n)i}(q)}{\partial q_n}v_i \end{bmatrix}$$

and is thus also given by the partial derivative of the inertia matrix. We thus only need to compute the partial derivative $\frac{\partial M(q)}{\partial q_i}$ once and use the result in the both in the first and second part of (2.57). This approach can be used to obtain the dynamic equations for an arbitrary n-link mechanism mounted on an AUV.

2.5 Spacecraft-Manipulator Systems

Spacecraft-manipulator systems are different from conventional earth-based manipulators in that they are placed in a free fall environment and that the base is not fixed (free-floating). In general there are three different cases that must be considered (Dubowsky and Papadopoulos, 1993). Firstly, if we have reaction jets available and use these to keep the spacecraft stationary we obtain a fixed spacecraft model which very much resembles the conventional fixed-based model. Secondly, if no actuation is used for the spacecraft we have a free-floating spacecraft with reduced fuel consumption at the expense of dynamic coupling between the spacecraft and the manipulator and a reduced workspace model. Finally, if the attitude, but not the position, of the spacecraft is actively controlled, we have a constrained spacecraft. We note that for free-floating spacecraft the center of mass (CM) of the spacecraft-manipulator system does not accelerate. However, when reaction jets or momentum wheels are used for control or other external forces are present, the center of mass is not constant in the orbit-fixed reference frame. The main challenge in modeling spacecraft-manipulator systems is that the base-fixed coordinate frame cannot simply be fixed in the orbit-fixed frame. There are two main approaches to deal with a floating base; the virtual manipulator approach (Vafa and Dubowsky, 1987) or the barycentric vector approach (Papadopoulos and Dubowsky, 1991).

2.5.1 State of the Art Spacecraft Dynamics

The attitude of a spacecraft is normally described by the Euler parameters, or unit quaternion. This is motivated by their properties as a nonsingular representation. We note that this is not the minimal representation, nor generalized coordinates, and thus not suited for the Lagrangian approach. Also, when transforming back to Euler angles from the unit quaternion representation a singularity is present for $\theta = \pm\frac{\pi}{2}$.

Any positive rotation ψ about a fixed unit vector n can be represented by the four-tuple

$$Q = \begin{bmatrix}\eta \\ \epsilon\end{bmatrix}, \tag{2.70}$$

where $\eta \in \mathbb{R}$ is known as the scalar part and $\epsilon \in \mathbb{R}^3$ as the vector part. $Q(\psi, n)$ is written in terms of ψ and n by

$$\eta = \cos(\frac{\psi}{2}), \qquad \epsilon = \sin(\frac{\psi}{2})n. \tag{2.71}$$

The kinematic differential equations can now be given by

$$\dot{\eta} = -\frac{1}{2}\epsilon^{\mathsf{T}}\omega_{0b}^{0} \tag{2.72}$$

$$\dot{\epsilon} = \frac{1}{2}(\eta I_b + \hat{\epsilon})\omega_{0b}^{0} \tag{2.73}$$

where ω_{0b}^{0} is the angular velocity of the body frame with respect to the orbit frame and I_b is the spacecraft inertia matrix. The attitude dynamics are given by (Hughes, 2002)

$$I_b\dot{\omega}_{0b}^{0} + \hat{\omega}_{0b}^{0}I_b\omega_{0b}^{0} = \tau. \tag{2.74}$$

2.5.2 State of the Art Spacecraft-Manipulator Dynamics

The equations of motion of a spacecraft-manipulator system can be written as (Egeland and Sagli, 1993)

$$M(Q)\dot{v} + C(Q, v)v = \tau. \tag{2.75}$$

Here, $v = \begin{bmatrix}\dot{r}_0^{\mathsf{T}} & (\omega_{0b}^{0})^{\mathsf{T}} & \dot{q}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$ where r_0 is the position of the center of mass of the vehicle, ω_{0b}^{0} the angular velocity of the vehicle and q is the joint position of the manipulator.

Alternatively we can use the center of mass of the whole system to represent the translational motion. Then $v = \begin{bmatrix}\dot{r}_{cm}^{\mathsf{T}} & (\omega_{0b}^{0})^{\mathsf{T}} & \dot{q}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$ where $\dot{r}_{cm}$ is the linear velocity of the center of mass of the vehicle-manipulator system. This is decoupled from the angular velocity ω_{0b}^{0} and the inertia matrix of a free-flying spacecraft-manipulator system can be written as (Dubowsky and Papadopoulos, 1993)

$$M = \begin{bmatrix} mI & 0 & 0 \\ 0 & M_{\omega\omega} & M_{q\omega}^{\mathsf{T}} \\ 0 & M_{q\omega} & M_{qq} \end{bmatrix} \tag{2.76}$$

where m is the total mass of the system. The Euler angle rates $\dot{\Theta}_{0b}$ relate to ω_{0b}^{0} by

$$\dot{\Theta}_{0b} = T_{\Theta_{0b}}(\Theta_{0b})\omega_{0b}^{0}. \tag{2.77}$$

Again $T_{\Theta_{0b}}(\Theta_{0b})$ is singular at isolated points. The control torques are given by $\tau = \begin{bmatrix}\tau_v^{\mathsf{T}} & \tau_\omega^{\mathsf{T}} & \tau_q^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$ where τ_v is the spacecraft forces generated by thrusters, τ_ω is the spacecraft moments generated by thrusters, momentum gyros or reaction wheels, and τ_q is the manipulator torques.

Other models are also available depending on the actuators available to control the spacecraft. In the case where $\tau_v, \tau_w \neq 0$ (free-flying space robots) the center of mass of the system is not constant, but described by the variable r_{cm} of Equation (2.75) if we let $v = \begin{bmatrix}\dot{r}_{cm}^{\mathsf{T}} & (\omega_{0b}^{0})^{\mathsf{T}} & \dot{q}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$. If no external forces act on the system and the spacecraft is

not actuated with thrusters, the center of mass does not accelerate, i.e., the system linear momentum is constant and $\dot{r}_{cm} = 0$. This can be used to simplify the equations to an n-dimensional system with inertia matrix $M_r = M_{qq} - M_{q\omega}M_{\omega\omega}^{-1}M_{q\omega}^{\mathsf{T}}$ and we get the reduced system by eliminating ω (Dubowsky and Papadopoulos, 1993; Papadopoulos and Dubowsky, 1991)

$$M_r(Q)\ddot{q} + C_r(Q, v)\dot{q} = \tau_q. \tag{2.78}$$

The attitude of the spacecraft is then found from

$$\omega = -M_{\omega\omega}^{-1}M_{q\omega}^{\mathsf{T}}\dot{q}. \tag{2.79}$$

The dynamic coupling between the manipulator and the spacecraft complicates the modeling and control of such systems. One way to deal with this is to derive a fixed-based manipulator with the same kinematic and dynamic properties as the free-floating spacecraft-manipulator system. The dynamically equivalent manipulator (DEM) (Liang et al., 1998; Parlaktuna and Ozkan, 2004) is a fixed-base manipulator with the base fixed in the center of mass of the space manipulator. Here, space manipulator refers to both the satellite and the manipulator. When no external forces are present, the center of mass does not move and the end-effector of this manipulator is thus given in the inertial frame. It can be shown that a given sequence of actuator torques acting on the DEM will produce the same joint trajectories for the space manipulator as for the DEM.

The dynamic equations of the free-floating space manipulator can be derived from from Lagrange's equations. We assume that all the joints are stiff and a free fall environment. The Lagrangian of the space manipulator is then given by the kinetic energy only, i.e.,

$$\mathcal{T} := \sum_{i=b}^{n+1}\left[\frac{1}{2}\dot{\rho}_i^{\mathsf{T}}m_i\dot{\rho}_i + \frac{1}{2}\omega_i^{\mathsf{T}}R_{0i}I_iR_{0i}^{\mathsf{T}}\omega_i\right] \tag{2.80}$$

for both the spacecraft and the links, which is different from Equation (2.12) in that the inertia matrix depends on the configuration of both the spacecraft and the joints. m_i is the total mass of link i and ρ_i is the distance from the center of mass of the system to the center of mass of link i.

Similarly, we can define a fixed-based manipulator with a spherical first joint and kinetic energy

$$\mathcal{T}' := \sum_{i=1}^{n+1}\left[\frac{1}{2}v_i^{\mathsf{T}}m_i'v_i + \frac{1}{2}(\omega_i')^{\mathsf{T}}R_{0i}'I_i'(R_{0i}')^{\mathsf{T}}\omega_i'\right] \tag{2.81}$$

where v_i is the velocity of link i with respect to the base. It can be shown that the kinematic and dynamic parameters of the space manipulator can be mapped to the DEM by (Liang

et al., 1998; Parlaktuna and Ozkan, 2004)

$$
\begin{aligned}
m_i' &= m_i \frac{\left(\sum_{k=1}^{n+1} m_k\right)^2}{\sum_{k=1}^{i-1} m_k \sum_{k=1}^{i} m_k}, \ i = 2 \dots n+1, \\
I_i' &= I_i, \ i = 1 \dots n+1, \\
W_1 &= \frac{R_1 m_1}{\sum_{k=1}^{n+1} m_k}, \\
W_i &= R_i \left(\frac{\sum_{k=1}^{i} m_k}{\sum_{k=1}^{n+1} m_k} \right) + L_i \left(\frac{\sum_{k=1}^{i-1} m_k}{\sum_{k=1}^{n+1} m_k} \right), \ i = 2 \dots n+1, \\
l_{c1} &= 0, \\
l_{ci} &= L_i \left(\frac{\sum_{k=1}^{i-1} m_k}{\sum_{k=1}^{n+1} m_k} \right), \ i = 2 \dots n+1,
\end{aligned} \tag{2.82}
$$

where the vector W_i connecting joint i with joint $i+1$ of the DEM is given by R_i and L_i of the space manipulator where R_i is the vector connecting the center of mass of link i and joint $i+1$ and L_i is the vector connecting joint i with the center of mass of link i. l_{ci} is the vector connecting joint i and the center of mass of joint i in the DEM. We refer to Liang et al. (1998) and Parlaktuna and Ozkan (2004) for details.

2.5.3 The Proposed Approach - $SE(3)$

As for the AUV, the configuration space of a spacecraft can be described by the matrix Lie group $SE(3)$ with respect to an orbit-fixed frame. The dynamic equations can be written as

$$
\begin{bmatrix} M_{VV} & M_{qV}^{\mathsf{T}} \\ M_{qV} & M_{qq} \end{bmatrix} \begin{bmatrix} \dot{V}_{0b}^b \\ \ddot{q} \end{bmatrix} + \begin{bmatrix} C_{VV} & C_{Vq} \\ C_{qV} & C_{qq} \end{bmatrix} \begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \tau_V \\ \tau_q \end{bmatrix} \tag{2.83}
$$

where

$$
C(Q,v) = \sum_{k=1}^{n} \frac{\partial M}{\partial q_k} \dot{q}_k - \frac{1}{2} \begin{bmatrix} 2\,\mathrm{ad}_{(M(q)v)_V}^{\mathsf{T}} & 0 \\ \frac{\partial^{\mathsf{T}}}{\partial q} \left(\begin{bmatrix} M_{VV} & M_{qV}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix} \right) & \frac{\partial^{\mathsf{T}}}{\partial q} \left(\begin{bmatrix} M_{qV} & M_{qq} \end{bmatrix} \begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix} \right) \end{bmatrix}. \tag{2.84}
$$

This can be used both for actuated and unactuated spacecraft.

For free-floating spacecraft we have $\tau_V = 0$ and we can simplify the dynamics substantially by re-writing the mass matrix as

$$
M_r = M_{qq} - M_{qV} M_{VV} M_{qV}^{\mathsf{T}}. \tag{2.85}
$$

The Coriolis matrix is then found by

$$
C_r(Q,v) = \sum_{k=1}^{n} \frac{\partial M_r}{\partial q_k} \dot{q}_k - \frac{1}{2} \frac{\partial^{\mathsf{T}}}{\partial q} (M_r v) \tag{2.86}
$$

with M_r given as in (2.85) and the dynamics are described by

$$M_r\ddot{q} + C_r^{\mathsf{T}}\dot{q} = \tau_q. \tag{2.87}$$

When $\ddot{q}$ and $\dot{q}$ are known, the base velocity vector can be found by

$$M_{VV}\dot{V}_{0b}^b + C_{VV}V_{0b}^b = -(M_{qV}^{\mathsf{T}}\ddot{q} + C_{Vq}\dot{q}). \tag{2.88}$$

This can be done either by projecting g_{0b} onto the allowed configuration space $SE(3)$ (McLachlan and Quispel, 2006) or by using structure-preserving integration methods (Munthe-Kaas, 1998). As these equations are based on the singularity-free dynamics (2.83) these are also singularity-free with the state variables $Q = \{g \in SE(3), q \in \mathbb{R}^n\}$ and $v = \begin{bmatrix}(V_{0b}^b)^{\mathsf{T}} & \dot{q}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^{6+n}$.

2.5.4 The Proposed Approach: The Dynamically Equivalent Manipulator - $SO(3)$

In this section we reformulate the dynamic equations of a space manipulator and its dynamically equivalent manipulator using the proposed framework. This removes the singularities in the representation, but is otherwise similar. Assume no spacecraft actuation, i.e., $\dot{r}_{cm} = 0$. Then the kinetic energy of link i of the space manipulator is given by

$$\begin{aligned}
\mathcal{T}_i &= \frac{1}{2}\left(V_{0i}^i\right)^{\mathsf{T}} I_i V_{0i}^i \\
&= \frac{1}{2}\left((\tilde{V}_{0b}^b)^{\mathsf{T}}H^{\mathsf{T}} + \dot{q}^{\mathsf{T}}J_i(q)^{\mathsf{T}}\right)\mathrm{Ad}_{g_{ib}}^{\mathsf{T}} I_i \mathrm{Ad}_{g_{ib}}\left(H\tilde{V}_{0b}^b + J_i(q)\dot{q}\right) \\
&= \frac{1}{2}\left((\omega_{0b}^0)^{\mathsf{T}}H^{\mathsf{T}} + \dot{q}^{\mathsf{T}}J_i(q)^{\mathsf{T}}\right)\mathrm{Ad}_{g_{ib}}^{\mathsf{T}} I_i \mathrm{Ad}_{g_{ib}}\left(H\omega_{0b}^0 + J_i(q)\dot{q}\right) \\
&= \frac{1}{2}\begin{bmatrix}(\omega_{0b}^0)^{\mathsf{T}} & \dot{q}^{\mathsf{T}}\end{bmatrix} M_i(q)\begin{bmatrix}\omega_{0b}^0 \\ \dot{q}\end{bmatrix} = \frac{1}{2}v^{\mathsf{T}}M_i(q)v
\end{aligned} \tag{2.89}$$

where

$$M_i(q) := \begin{bmatrix}H^{\mathsf{T}}\mathrm{Ad}_{g_{ib}}^{\mathsf{T}} I_i \mathrm{Ad}_{g_{ib}} H & H^{\mathsf{T}}\mathrm{Ad}_{g_{ib}}^{\mathsf{T}} I_i \mathrm{Ad}_{g_{ib}} J_i \\ J_i^{\mathsf{T}}\mathrm{Ad}_{g_{ib}}^{\mathsf{T}} I_i \mathrm{Ad}_{g_{ib}} H & J_i^{\mathsf{T}}\mathrm{Ad}_{g_{ib}}^{\mathsf{T}} I_i \mathrm{Ad}_{g_{ib}} J_i\end{bmatrix} \tag{2.90}$$

and the inertia matrix is given by substituting this into (2.13) and H given as in (2.28). The configuration space is then given by $Q = \{R_{0b}, q\}$.

Similarly, we can define a fixed-based manipulator with a spherical first joint, also with configuration space $SO(3)$. The corresponding inertia matrices are then given by

$$M_i'(q) := \begin{bmatrix}H^{\mathsf{T}}\mathrm{Ad}_{g_{ib}'}^{\mathsf{T}} I_i' \mathrm{Ad}_{g_{ib}'} H & H^{\mathsf{T}}\mathrm{Ad}_{g_{ib}'}^{\mathsf{T}} I_i' \mathrm{Ad}_{g_{ib}'} J_i' \\ (J_i')^{\mathsf{T}}\mathrm{Ad}_{g_{ib}'}^{\mathsf{T}} I_i' \mathrm{Ad}_{g_{ib}'} H & (J_i')^{\mathsf{T}}\mathrm{Ad}_{g_{ib}'}^{\mathsf{T}} I_i' \mathrm{Ad}_{g_{ib}'} J_i'\end{bmatrix} \tag{2.91}$$

where I_i' and the kinematic relations used to compute R_{0i}' and J_i' are found from (2.82). Thus, we have $\tilde{V}_{0b}^b = \tilde{V}_{0b}^{\prime b}$ as required. The spacecraft inertia matrix is given by

$$I_b = \begin{bmatrix}J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z\end{bmatrix} \tag{2.92}$$

which also represents the inertial properties of the spherical base link. The Coriolis matrix then becomes (following the mathematics of (2.27-2.31))

$$C'(Q,v) = \sum_{k=1}^{n} \frac{\partial M'}{\partial q_k}\dot{q}_k \tag{2.93}$$
$$- \frac{1}{2}\begin{bmatrix} -2(\widehat{M'(q)v})_{\tilde{V}} & 0 \\ \frac{\partial^\mathsf{T}}{\partial q}\left(\begin{bmatrix} M'_{VV} & (M')^\mathsf{T}_{qV}\end{bmatrix}\begin{bmatrix} V^b_{0b} \\ \dot{q}\end{bmatrix}\right) & \frac{\partial^\mathsf{T}}{\partial q}\left(\begin{bmatrix} M'_{qV} & (M')^\mathsf{T}_{qq}\end{bmatrix}\begin{bmatrix} V^b_{0b} \\ \dot{q}\end{bmatrix}\right)\end{bmatrix}$$

where $(M'(q)v)_{\tilde{V}}$ is the vector of the first three entries of the vector $M'(q)v$ (corresponding to $\tilde{V}^b_{0b} = \omega^0_{0b}$). The specific computations of the inertia and Coriolis matrices are performed in the same way as for the AUV (see Section 2.4.3) except from the partial derivatives of the inertia matrices which now depend on the selection matrix H. This is shown in Section 2.5.4.

The dynamic equations can now be written as

$$M'(q)\dot{v} + C'(Q,v)v = \tau. \tag{2.94}$$

Here, $v = \begin{bmatrix}(\omega^0_{0b})^\mathsf{T} & \dot{q}^\mathsf{T}\end{bmatrix}^\mathsf{T}$ where ω^0_{0b} is the velocity state of the passive spherical base joint of the DEM (and thus also the spacecraft) and $\dot{q}$ the velocity state of the manipulator of the DEM (and the space manipulator), and $Q = \{R_{0b}, q\}$ where $R_{0b} \in SO(3)$ determines the configuration of the spherical joint/spacecraft and q the configuration of the manipulators of the DEM and space manipulator. We note that the singularity that normally arises when using the Euler angles is eliminated and the state space (Q, v) is valid globally.

Most importantly, we can now use this fixed-base DEM for simulation and control of the space manipulator. Similar to the conventional approach, the DEM described by (2.94) have the same kinetic and dynamic properties as the space manipulator and if the same actuator torques $\tau(t) = \tau'(t)$, $\forall t$ are applied on both the DEM and the space manipulator, this will produce the same joint trajectory $q(t) = q'(t)$ for $\forall t \in [t_0, \infty]$ if $q(t_0) = q'(t_0)$.

Computing the Partial derivatives of $M(q_1, \ldots, q_n)$

The partial derivatives of the inertia matrix with respect to $q_1, \ldots, q_n$ are computed by

$$\frac{\partial M(q_1,\ldots,q_n)}{\partial q_k} = \sum_{i=k}^{n}\left(\begin{bmatrix} H^\mathsf{T} \\ J_i^\mathsf{T}\end{bmatrix}\left[\frac{\partial^\mathsf{T}\operatorname{Ad}_{g_{ib}}}{\partial q_k} I_i \operatorname{Ad}_{g_{ib}} + \operatorname{Ad}^\mathsf{T}_{g_{ib}} I_i \frac{\partial \operatorname{Ad}_{g_{ib}}}{\partial q_k}\right]\begin{bmatrix} H & J_i\end{bmatrix}\right)$$

$$+ \sum_{i=k+1}^{n}\begin{bmatrix} 0_{m\times m} & H^\mathsf{T}\operatorname{Ad}^\mathsf{T}_{g_{ib}} I_i \operatorname{Ad}_{g_{ib}} \frac{\partial J_i}{\partial q_k} \\ \frac{\partial^\mathsf{T} J_i}{\partial q_k}\operatorname{Ad}^\mathsf{T}_{g_{ib}} I_i \operatorname{Ad}_{g_{ib}} H & \frac{\partial^\mathsf{T} J_i}{\partial q_k}\operatorname{Ad}^\mathsf{T}_{g_{ib}} I_i \operatorname{Ad}_{g_{ib}} J_i + J_i^\mathsf{T}\operatorname{Ad}^\mathsf{T}_{g_{ib}} I_i \operatorname{Ad}_{g_{ib}} \frac{\partial J_i}{\partial q_k}\end{bmatrix} \tag{2.95}$$

which only differs from (2.59) in that the identity matrix I is substituted by H and H^T in the first part and we multiply by H and H^T to get the right dimensions in the second part.

2.6 Ground Vehicle-Manipulator Systems

We now consider a ground vehicle with no non-holonomic constraints. The configuration space can be described by the matrix Lie group $SE(2)$. The velocity state is thus fully determined by only three variables and we choose H so that

$$V_{0b}^b = H\tilde{V}_{0b}^b \tag{2.96}$$

with

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\mathsf{T}. \tag{2.97}$$

For Euclidean joints Equation (2.19) simplifies to

$$\tilde{V}_{0b}^b = \dot{\phi}_V. \tag{2.98}$$

S is thus given by the identity matrix, the partial derivatives of S vanish and we get

$$C_{VV}(Q, v) = \sum_{k=1}^{6} \frac{\partial M_{VV}}{\partial q_k} \dot{q}_k. \tag{2.99}$$

The inertia matrix

$$I = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J_z \end{bmatrix} \tag{2.100}$$

then determines the dynamic equations.

If non-holonomic constraints are present, such as for wheeled mechanisms, we get the selection matrix

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\mathsf{T} \tag{2.101}$$

and velocity state $\tilde{V}_{0b}^b = \left[\begin{smallmatrix} v_x \\ \omega_z \end{smallmatrix}\right]$. The dynamics are then found by substituting H and $\tilde{V}_{0b}^b$ into the formalism presented in Section 2.3.

2.7 A Simple Example

Consider the general structure of the equations for a mechanism with one joint with joint variable q_1 mounted on a vehicle with configuration space $SE(3)$. We can write its inertia matrix as follows

$$M(q_1) = \begin{bmatrix} I_b + \mathrm{Ad}_{g_{1b}}^\mathsf{T} I_1 \mathrm{Ad}_{g_{1b}} & \mathrm{Ad}_{g_{1b}}^\mathsf{T} I_1 \mathrm{Ad}_{g_{1b}} X_1 \\ X_1^\mathsf{T} \mathrm{Ad}_{g_{1b}}^\mathsf{T} I_1 \mathrm{Ad}_{g_{1b}} & X_1^\mathsf{T} \mathrm{Ad}_{g_{1b}}^\mathsf{T} I_1 \mathrm{Ad}_{g_{1b}} X_1 \end{bmatrix}. \tag{2.102}$$

Its partial derivative with respect to q is a single matrix

$$\frac{\partial M(q_1)}{\partial q_1} = \begin{bmatrix} I \\ X_1^\mathsf{T} \end{bmatrix} \left[\frac{\partial^\mathsf{T} \mathrm{Ad}_{g_{1b}}}{\partial q_1} I_1 \mathrm{Ad}_{g_{1b}} + \mathrm{Ad}_{g_{1b}}^\mathsf{T} I_1 \frac{\partial \mathrm{Ad}_{g_{1b}}}{\partial q_1} \right] \begin{bmatrix} I & X_1 \end{bmatrix} \tag{2.103}$$

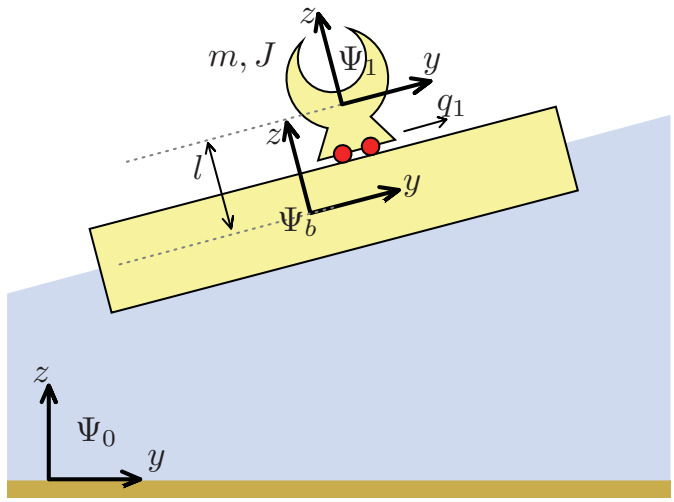


Figure 2.2: One-link robot with a prismatic joint attached to a non-inertial base with configuration space $SE(3)$.

with

$$\frac{\partial g_{1b}}{\partial q_1} = -g_{1b}\hat{X}_1 g_{bb} = -g_{1b}\hat{X}_1. \tag{2.104}$$

Note that the Jacobian matrix is constant and hence no partial derivatives are taken.

Consider as an example the robot in Figure 2.2 with a single prismatic joint. We can write the Jacobian as $J_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$ and the inertia matrix as

$$M(q) = \begin{bmatrix} M_b & 0 & 0 & 0 & ml & -mq_1 & 0 \\ 0 & M_b & 0 & -ml & 0 & 0 & m \\ 0 & 0 & M_b & mq_1 & 0 & 0 & 0 \\ 0 & -ml & mq_1 & J_{t,x} + ml^2 + mq_1^2 & 0 & 0 & -ml \\ ml & 0 & 0 & 0 & J_{t,y} + ml^2 & -mlq_1 & 0 \\ -mq_1 & 0 & 0 & 0 & -mlq_1 & J_{t,z} + mq_1^2 & 0 \\ 0 & m & 0 & -ml & 0 & 0 & m \end{bmatrix} \tag{2.105}$$

where $M_b = m_b + m$ and $J_{t,x} = J_{b,x} + J_x$, etc. Assume we are interested in the dynamics of the prismatic joint. This is given by the last row of the inertia and Coriolis matrix. The Coriolis matrix is given by (2.26) where the first part is zero and the second part gives

$$C(q, V_{0b}^b, \dot{q}) = \begin{bmatrix} * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ \frac{m}{2}\omega_z & 0 & -\frac{m}{2}\omega_x & -\frac{m}{2}v_z - mq_1\omega_x & \frac{m}{2}l\omega_z & \frac{m}{2}(v_x + l\omega_y) - mq_1\omega_z & 0 \end{bmatrix}. \tag{2.106}$$

The last row here is given by multiplying the $\frac{\partial M(q_1)}{\partial q_1} \in \mathbb{R}^{7\times 7}$ with the vector $v = \begin{bmatrix}(V_{0b}^b)^\mathsf{T} & \dot{q}_1\end{bmatrix}^\mathsf{T}$. Using these expressions, we can write the dynamics of the prismatic joint due to the motion of the vehicle as

$$\begin{bmatrix}M_{qV} & M_{qq}\end{bmatrix}\begin{bmatrix}\dot{V}_{0b}^b \\ \ddot{q}\end{bmatrix} + \begin{bmatrix}C_{qV} & C_{qq}^\mathsf{T}\end{bmatrix}\begin{bmatrix}V_{0b}^b \\ \dot{q}\end{bmatrix} = \tau$$

$$m\ddot{q}_1 + m\dot{v}_y - ml\dot{\omega}_x + \frac{m}{2}\omega_z v_x - \frac{m}{2}\omega_x v_z$$

$$-\frac{m}{2}v_z\omega_x - mq_1\omega_x^2 - \frac{m}{2}(v_x + l\omega_y)\omega_z - mq_1\omega_z^2 = \tau$$

$$\ddot{q}_1 + \dot{v}_y - l\dot{\omega}_x + (v_x + l\omega_y)\omega_z - v_z\omega_x - q_1\omega_x^2 - q_1\omega_z^2 = \frac{\tau}{m}. \tag{2.107}$$

Similarly, if we consider a single rigid body in $SE(3)$ the inertia matrix becomes (dropping the subscript b)

$$M = \begin{bmatrix}m & 0 & 0 & 0 & 0 & 0\\ 0 & m & 0 & 0 & 0 & 0\\ 0 & 0 & m & 0 & 0 & 0\\ 0 & 0 & 0 & J_x & 0 & 0\\ 0 & 0 & 0 & 0 & J_y & 0\\ 0 & 0 & 0 & 0 & 0 & J_z\end{bmatrix} \tag{2.108}$$

and when computing the Coriolis matrix we note that the first part of (2.26) is zero and the second part is given by $\mathrm{ad}_{(Mv)}^\mathsf{T}$ and the Coriolis matrix is thus given by

$$C(q) = \begin{bmatrix}0 & -J_z\omega_z & J_y\omega_y & 0 & 0 & 0\\ J_z\omega_z & 0 & -J_x\omega_x & 0 & 0 & 0\\ -J_y\omega_y & J_x\omega_x & 0 & 0 & 0 & 0\\ 0 & -mv_z & mv_y & 0 & -J_z\omega_z & J_y\omega_y\\ mv_z & 0 & -mv_x & J_z\omega_z & 0 & -J_x\omega_x\\ -mv_y & mv_x & 0 & -J_y\omega_y & J_x\omega_x & 0\end{bmatrix} \tag{2.109}$$

which we recognize as Kirchhoff's equations. Kirchhoff's equations are, however, valid for systems with only kinetic energy.

There are many ways for computing the Coriolis matrix for rigid bodies. One commonly found formulation in ship modeling is

$$C(q) = -\begin{bmatrix}0 & \widehat{M_{11}\nu_1} + \widehat{M_{12}\nu_2}\\ \widehat{M_{11}\nu_1} + \widehat{M_{12}\nu_2} & \widehat{M_{21}\nu_1} + \widehat{M_{22}\nu_2}\end{bmatrix} \tag{2.110}$$

and the dynamics are given by (2.39) and (2.40). The expression in (2.110) can also be reformulated to the form of (2.109). We note that using this approach we end up with the transformation in (2.39) which singularity prone.

2.8 Conclusions

In this paper the dynamic equations of vehicle-manipulator systems are derived based on Lagrange's equations. The main contribution is to close the gap between manipulator dy-

namics which are normally derived based on the Lagrangian approach and vehicle dynamics which are normally derived using other approaches in order to avoid singularities. The proposed framework allows us to derive the dynamics of vehicles with a Lie group topology using a minimal, singularity-free representation based on Lagrange's equations which naturally extends to include also the manipulator dynamics. The globally valid vehicle-manipulator dynamics are thus derived for the first time using the proposed framework. Several examples of how to derive the dynamics for different vehicles, such as spacecraft, AUVs, and ground vehicles are shown to illustrate the simple analytical form of the final equations.

2.9 Appendix

2.9.1 Partial Derivatives of Ad_g - By Direct Computation

The partial derivative of $\mathrm{Ad}_{g_{ij}}$ with respect to q_k when $i < k \leq j$ can be written as

$$
\begin{aligned}
&\frac{\partial \,\mathrm{Ad}_{g_{ij}}}{\partial q_k} = \mathrm{Ad}_{g_{i(k-1)}} \frac{\partial \,\mathrm{Ad}_{g_{(k-1)k}}}{\partial q_k} \mathrm{Ad}_{g_{kj}} \\
&= \begin{bmatrix} R_{i(k-1)} & \hat{p}_{i(k-1)} R_{i(k-1)} \\ 0 & R_{i(k-1)} \end{bmatrix} \begin{bmatrix} \frac{\partial R_{(k-1)k}}{\partial q_k} & \frac{\hat{p}_{(k-1)k}}{\partial q_k} R_{(k-1)k} + \hat{p}_{(k-1)k} \frac{\partial R_{(k-1)k}}{\partial q_k} \\ 0 & \frac{\partial R_{(k-1)k}}{\partial q_k} \end{bmatrix} \begin{bmatrix} R_{kj} & \hat{p}_{kj} R_{kj} \\ 0 & R_{kj} \end{bmatrix} \\
&= \begin{bmatrix} R_{i(k-1)} \frac{\partial R_{(k-1)k}}{\partial q_k} R_{kj} & \begin{bmatrix} R_{i(k-1)} \frac{\partial R_{(k-1)k}}{\partial q_k} \hat{p}_{kj} R_{kj} + R_{i(k-1)} \frac{\hat{p}_{(k-1)k}}{\partial q_k} R_{(k-1)j} + \\ R_{i(k-1)} \hat{p}_{(k-1)k} \frac{\partial R_{(k-1)k}}{\partial q_k} R_{kj} + \hat{p}_{i(k-1)} R_{i(k-1)} \frac{\partial R_{(k-1)k}}{\partial q_k} R_{kj} \end{bmatrix} \\ 0 & R_{i(k-1)} \frac{\partial R_{(k-1)k}}{\partial q_k} R_{kj} \end{bmatrix} \\
&= \begin{bmatrix} R_{i(k-1)} \hat{X}_\omega R_{(k-1)k} R_{kj} & \begin{bmatrix} R_{i(k-1)} \hat{X}_\omega R_{(k-1)k} \hat{p}_{kj} R_{kj} + R_{i(k-1)} (\widehat{(\hat{X}_\omega p_{(k-1)k})} + \hat{X}_v) R_{(k-1)j} \\ + R_{i(k-1)} \hat{p}_{(k-1)k} \hat{X}_\omega R_{(k-1)k} R_{kj} + \hat{p}_{i(k-1)} R_{i(k-1)} \hat{X}_\omega R_{(k-1)k} R_{kj} \end{bmatrix} \\ 0 & R_{i(k-1)} \hat{X}_\omega R_{(k-1)k} R_{kj} \end{bmatrix} \\
&= \begin{bmatrix} R_{i(k-1)} \hat{X}_\omega R_{(k-1)j} & \begin{bmatrix} R_{i(k-1)} \hat{X}_\omega R_{(k-1)k} \hat{p}_{kj} R_{kj} + R_{i(k-1)} (\hat{X}_\omega \widehat{p_{(k-1)k}}) R_{(k-1)j} + \\ R_{i(k-1)} \hat{X}_v R_{(k-1)j} + R_{i(k-1)} \hat{p}_{(k-1)k} \hat{X}_\omega R_{(k-1)j} + \hat{p}_{i(k-1)} R_{i(k-1)} \hat{X}_\omega R_{(k-1)j} \end{bmatrix} \\ 0 & R_{i(k-1)} \hat{X}_\omega R_{(k-1)j} \end{bmatrix} \\
&= \begin{bmatrix} R_{i(k-1)} \hat{X}_\omega R_{(k-1)j} & R_{i(k-1)} \hat{X}_v R_{(k-1)j} + R_{i(k-1)} \hat{X}_\omega \hat{p}_{(k-1)j} R_{(k-1)j} + \hat{p}_{i(k-1)} R_{i(k-1)} \hat{X}_\omega R_{(k-1)j} \\ 0 & R_{i(k-1)} \hat{X}_\omega R_{(k-1)j} \end{bmatrix} \\
&= \begin{bmatrix} R_{i(k-1)} \hat{X}_\omega & R_{i(k-1)} \hat{X}_v + \hat{p}_{i(k-1)} R_{i(k-1)} \hat{X}_\omega \\ 0 & R_{i(k-1)} \hat{X}_\omega \end{bmatrix} \begin{bmatrix} R_{(k-1)j} & \hat{p}_{(k-1)j} R_{(k-1)j} \\ 0 & R_{(k-1)j} \end{bmatrix} \\
&= \begin{bmatrix} R_{i(k-1)} & \hat{p}_{i(k-1)} R_{i(k-1)} \\ 0 & R_{i(k-1)} \end{bmatrix} \begin{bmatrix} \hat{X}_\omega & \hat{X}_v \\ 0 & \hat{X}_\omega \end{bmatrix} \begin{bmatrix} R_{(k-1)j} & \hat{p}_{(k-1)j} R_{(k-1)j} \\ 0 & R_{(k-1)j} \end{bmatrix} \\
&= \mathrm{Ad}_{g_{i(k-1)}} \,\mathrm{ad}_{X_k} \,\mathrm{Ad}_{g_{(k-1)j}}
\end{aligned}
\tag{2.111}
$$

where we have used that

$$\hat{a}\hat{b} = \widehat{(\hat{a}b)} + \hat{b}\hat{a}, \tag{2.112}$$

and

$$\hat{p}_{(k-1)j} = \widehat{(R_{(k-1)k} p_{kj})} + \hat{p}_{(k-1)k}. \tag{2.113}$$

The proof when $j < k \leq i$ follows the same approach.

2.9.2 Partial Derivatives of the Mass Matrix for Joints with Non- Constant Twist

For a non-constant twist X_k, we get the following expression for the partial derivatives of the inertia matrix

$$\frac{\partial M(q_1,\ldots,q_n)}{\partial q_k} = \sum_{i=k}^{n} \left(\begin{bmatrix} H^\mathsf{T} \\ J_i^\mathsf{T} \end{bmatrix} \left[\tfrac{\partial^\mathsf{T} \mathrm{Ad}_{g_{ib}}}{\partial q_k} I_i \, \mathrm{Ad}_{g_{ib}} + \mathrm{Ad}_{g_{ib}}^\mathsf{T} \, I_i \tfrac{\partial \, \mathrm{Ad}_{g_{ib}}}{\partial q_k} \right] \begin{bmatrix} H & J_i \end{bmatrix} \right)$$

$$+ \sum_{i=k}^{n} \begin{bmatrix} 0_{m\times m} & H^\mathsf{T} \, \mathrm{Ad}_{g_{ib}}^\mathsf{T} \, I_i \, \mathrm{Ad}_{g_{ib}} \, \tfrac{\partial J_i}{\partial q_k} \\ \tfrac{\partial^\mathsf{T} J_i}{\partial q_k} \, \mathrm{Ad}_{g_{ib}}^\mathsf{T} \, I_i \, \mathrm{Ad}_{g_{ib}} \, H & \tfrac{\partial^\mathsf{T} J_i}{\partial q_k} \, \mathrm{Ad}_{g_{ib}}^\mathsf{T} \, I_i \, \mathrm{Ad}_{g_{ib}} \, J_i + J_i^\mathsf{T} \, \mathrm{Ad}_{g_{ib}}^\mathsf{T} \, I_i \, \mathrm{Ad}_{g_{ib}} \, \tfrac{\partial J_i}{\partial q_k} \end{bmatrix} \tag{2.114}$$

where the only difference from the constant twist expression (2.95) is that the summing starts from k and not $k+1$ in the last term and that the partial derivatives of the Jacobian are given by

$$\tfrac{\partial J_i}{\partial q_k} = \big[0 \quad \underbrace{\mathrm{Ad}_{g_{b(k-1)}} \tfrac{\partial}{\partial q_k} X_k(q_k)}_{\text{For non-constant twists only}} \quad \tfrac{\partial}{\partial q_k}(\mathrm{Ad}_{g_{bk}}) X_{k+1} \quad \cdots \quad \tfrac{\partial}{\partial q_k}(\mathrm{Ad}_{g_{b(i-1)}}) X_i(q_i) \quad 0 \big] \tag{2.115}$$

We still have that $q_k = \bar{q}_k + \phi$ and thus for a constant $\bar{q}_k$ we get $\dot{q}_k = \dot{\phi}_k$ so that the transformation from local to global coordinates for the manipulator is still given by $\dot{q} = S(q,\phi)\dot{\phi}$ with $S(q,\phi) = I$. Thus the expression for the Coriolis matrix does not change.

Chapter 3

Modeling and Motion Planning for Mechanisms on a Non-Inertial Base

3.1 Abstract

Robotic manipulators on ships and platforms suffer from large inertial forces due to the non-inertial motion of the ship or platform. When operating in high sea state, operation of such manipulators can be made more efficient and robust if these non-inertial effects are taken into account in the motion planning and control systems.

Motivated by this application, we present a rigorous and singularity-free formulation of the dynamics of a robotic manipulator mounted on a non-inertial base. We extend the classical dynamics equations for a serial manipulator to include the 6-DoF motion of the non-inertial base. Then, we show two examples of a 1-DoF and a 4-DoF manipulator to illustrate how these non-inertial effects can be taken into account in the motion planning.

3.2 Introduction

The use of unmanned and autonomous vehicles operating in hostile environments has shown both to be cost efficient and to protect humans from potentially dangerous situations. One such hostile environment is high sea state. We look into the case when a manipulator mounted on a ship or a platform is required to operate independently of the sea state. Large inertial forces may influence the manipulator and make the operation inaccurate, extremely energy demanding, or impossible due to torque limits. The inertial forces thus need to be taken into account in both the path planning and control of the robot.

Ships and oil platforms are expected to become increasingly unmanned in the future and hence the need for continuously operating robots for surveillance, maintenance, and operation will grow (Love et al., 2004; Kitarovic et al., 2005). All these tasks become

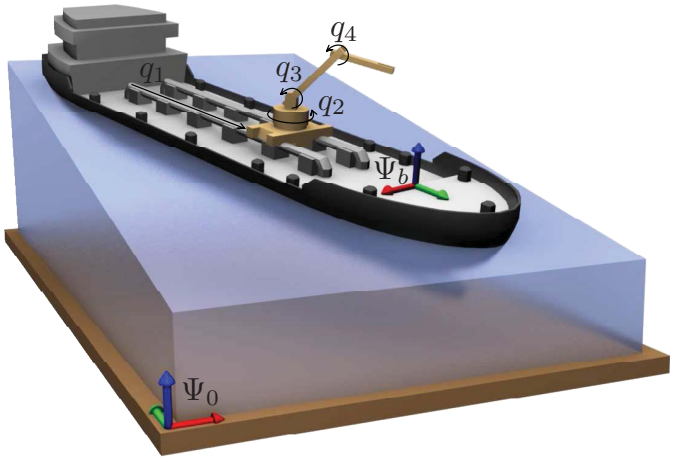


Figure 3.1: Model setup for a four-link robot attached to a non-inertial base with coordinate frame Ψ_b. Frame Ψ_0 denotes the inertial reference frame.

increasingly important in harsh environments such as high sea state. To be able to continue operation under these conditions, a good understanding is needed of the effects of the inertial forces due to the motion of the ship or platform. We therefore develop the dynamic equations of the robotic manipulator including these effects.

Research on several related topics can be found in literature. Love et al. (2004) addressed the impact of wave generated disturbances on the tracking control of a manipulator mounted on a ship based on the classical Lagrangian approach. Repetitive learning control was used and performance was improved for purely periodic motions, but no formal derivation of the dynamics equations was presented. The use of cable robots for loading and unloading cargo between two ships has also been addressed by Kitarovic et al. (2005) and Oh et al. (2005). In the Ampelmann project (Salzmann, 2007), a Stewart platform is mounted on a ship and is used to compensate for the motion of the ship by keeping the platform still with respect to the world frame. Lebans et al. (1997) give a cursory description of a telerobotic shipboard handling system, and Kosuge et al. (1992); Kajita and Kosuge (1997) addresses the control of robots floating on the water utilizing vehicle restoring forces. Other related research areas are macro/micro manipulators (Yoshikawa et al., 1996; Bowling and Khatib, 1997), underwater vehicle/manipulator systems (McMillan et al., 1995) and spacecraft/manipulator systems (Egeland and Sagli, 1993). Most previous work deals with robots mounted on a free-floating base. There is, however, an important difference between modeling a robot on a forced and a free-floating base. A forced base motion will add inertial forces to the dynamic equations that do not arise in free-floating case, such as spacecraft/manipulator systems and manipulators on small AUVs.

Our approach differs from previous work in that the dynamic equations are derived for rigid multibody systems including both Euclidean joints and generalized joints with configuration spaces different from $\mathbb{R}^n$. We follow the generalized Lagrangian approach presented in Duindam and Stramigioli (2007, 2008), which allows us to combine the Euclidean joints (the manipulator) and more general joints (the base), i.e. joints that can be

described by the Lie group $SE(3)$ or one of its ten subgroups. In our case, the transformation from the inertial frame to the base frame (the platform) is represented as a "free motion" joint described by the Lie group $SE(3)$. We also show through the examples how the base motion can be expressed as subgroups of $SE(3)$, in our case $SO(2)$.

For marine vessels in high sea state, very large inertial forces are added to the manipulator dynamics. To illustrate the effect of the inertial forces and how these appear in the robot equations, we look into the problem of finding the optimal trajectory in terms of actuator torque. There are many motivations for doing this. First of all the wear and tear on the manipulator is reduced and, for cooperative manipulation, the possibility of breaking an object manipulated by two robots is also reduced. Secondly, the solution is more energy efficient as the inertial forces will, if possible, contribute to the desired motion instead of working against it. The final motivation is that a good understanding of the effects of the inertial forces on the dynamic equations is essential in tasks that require high accuracy and the need to compensate for these effects (Love et al., 2004).

We assume that the motion of the free moving base is forced externally by forces unknown to us and that the pose, velocity, and acceleration of the base relative to the inertial world are known for all times. This means we also assume that the motion of the robot does not influence the motion of the base. The motion of the base will add inertial forces to the dynamics equations of the robot and the pose of the base will influence the gravitational forces acting on each link. Finally, we assume the robot to be an ideal rigid friction-less mechanism with purely torque driven actuators.

Given these assumptions, we consider the following two problems: first, we derive the dynamic equations describing the motion of the robot under the influence of the non-inertial base motion. Second, we consider the path planning problem of finding the trajectory between two given configurations for a given base motion. Intuitively, the optimal solution is the trajectory for which the inertial forces help accelerate and decelerate the robot as much as possible, such that little control torque is required.

3.3 Multibody Dynamics with a Non-Inertial Base

We extend the classical dynamics equations for a serial manipulator arm with 1-DoF joints to include the forced 6-DoF motion of the non-inertial base.

3.3.1 Manipulator Kinematics on a Non-Inertial Base

Consider the setup of Fig. 3.1 describing a general n-link robot manipulator arm attached to a moving base. Choose an inertial coordinate frame Ψ_0, a frame Ψ_b rigidly attached to the moving base, and n frames Ψ_i (not shown) attached to each link i at the center of mass with axes aligned with the principal directions of inertia. Finally, choose a vector $q \in \mathbb{R}^n$ that describes the configuration of the n joints.

Using standard notation (Murray et al., 1994), we can describe the pose of each frame Ψ_i relative to Ψ_0 as a homogeneous transformation matrix $g_{0i} \in SE(3)$ of the form

$$g_{0i} = \begin{bmatrix} R_{0i} & p_{0i} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{3.1}$$

with rotation matrix $R_{0i} \in SO(3)$ and translation vector $p_{0i} \in \mathbb{R}^3$. This pose can also be described using the vector of joint coordinates q as

$$g_{0i} = g_{0b} g_{bi} = g_{0b} g_{bi}(q) \tag{3.2}$$

The base pose g_{0b} and the joint positions q thus fully determine the configuration state of the robot.

In a similar way, the spatial velocity of each link can be expressed using twists (Murray et al., 1994):

$$V_{0i}^0 = \begin{bmatrix} v_{0i}^0 \\ \omega_{0i}^0 \end{bmatrix} = V_{0b}^0 + V_{bi}^0 = \mathrm{Ad}_{g_{0b}} \left(V_{0b}^b + J_i(q)\dot{q} \right) \tag{3.3}$$

where v_{0i}^0 and ω_{0i}^0 are the linear and angular velocities, respectively, of link i relative to the inertial frame, $J_i(q) \in \mathbb{R}^{6\times n}$ is the geometric Jacobian of link i relative to Ψ_b, the adjoint is defined as $\mathrm{Ad}_g := \left[\begin{smallmatrix} R & \hat{p}R \\ 0 & R \end{smallmatrix} \right] \in \mathbb{R}^{6\times 6}$, and $\hat{p} \in \mathbb{R}^{3\times 3}$ is the skew symmetric matrix such that $\hat{p}x = p \times x$ for all $p, x \in \mathbb{R}^3$. The velocity state is thus fully determined given the twist V_{0b}^b of the base and the joint velocities $\dot{q}$.

3.3.2 Manipulator Dynamics on a Non-Inertial Base

The previous section shows how the kinematics of the system can be naturally described in terms of the (global) state variables g_{0b}, q, V_{0b}^b, and $\dot{q}$. We now derive the dynamics equations for the system in terms of these state variables. We first assume the base to be free-moving under the influence of some prescribed external wrench F, and then restrict the base motion to be kinematically constrained.

To derive the dynamics of the complete mechanism (including the 6-DoF between Ψ_0 and Ψ_b), we follow the generalized Lagrangian method introduced by Duindam and Stramigioli (2007, 2008). This method gives the dynamics equations for a general mechanism described by a set $Q = \{Q_i\}$ of configuration states Q_i (not necessarily Euclidean), a vector v of velocity states $v_i \in \mathbb{R}^{n_i}$, and several mappings that describe the local Euclidean structure of the configuration states and their relation to the velocity states. More precisely, the neighborhood of every state $\bar{Q}_i$ is locally described by a set of Euclidean coordinates $\phi_i \in \mathbb{R}^{n_i}$ as $Q_i = Q_i(\bar{Q}_i, \phi_i)$ with $Q_i(\bar{Q}_i, 0) = \bar{Q}_i$, and there exist differentiable matrices S_i such that we can write $v_i = S_i(Q_i, \phi_i)\dot{\phi}_i$ for every Q_i.

Given a mechanism with coordinates formulated in this generalized form, we can write its kinetic energy as $U_k(Q, v) = \frac{1}{2} v^T M(Q) v$ with $M(Q)$ the inertia matrix in coordinates Q. The dynamics of this system then satisfy

$$M(Q)\dot{v} + \bar{C}(Q, v)v = \bar{\tau} \tag{3.4}$$

with τ the vector of gravitational, friction, and other external torques (collocated with v), and $\bar{C}(Q, v)$ the matrix describing Coriolis and centrifugal forces and given by

$$\begin{aligned} \bar{C}_{ij}(Q, v) := & \sum_{k,l} \left(\frac{\partial M_{ij}}{\partial \phi_k} S_{kl}^{-1} - \frac{1}{2} S_{ki}^{-1} \frac{\partial M_{jl}}{\partial \phi_k} \right) \Bigg|_{\phi=0} v_l \\ & + \sum_{k,l,m,s} \left(S_{mi}^{-1} \left(\frac{\partial S_{mj}}{\partial \phi_s} - \frac{\partial S_{ms}}{\partial \phi_j} \right) S_{sk}^{-1} M_{kl} \right) \Bigg|_{\phi=0} v_l \end{aligned} \tag{3.5}$$

More details and proofs can be found in references Duindam and Stramigioli (2007) and Duindam and Stramigioli (2008).

To apply this general result to systems of the form of Fig. 3.1, we write $Q = \{g_{0b}, q\}$ as the set of configuration states, and $v = \begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix}$ as the vector of velocity states. The local Euclidean structure for the state g_{0b} is given by exponential coordinates (Murray et al., 1994), while the state q is globally Euclidean of itself. Mathematically, we can express configurations (g_{0b}, q) around a fixed state $(\bar{g}_{0b}, \bar{q})$ as

$$g_{0b} = \bar{g}_{0b} \exp\left(\sum_{j=1}^{6} b_j (\phi_b)_j\right) \tag{3.6}$$

$$q_i = \bar{q}_i + \phi_i \quad \forall\, i \in \{1, \ldots, n\} \tag{3.7}$$

with b_j the standard basis elements of the Lie algebra $se(3)$. The corresponding matrices S_i can be collected in one block-diagonal matrix S given by (Duindam, 2006)

$$S(Q, \phi) = \begin{bmatrix} \left(I - \frac{1}{2}\operatorname{ad}_{\phi_b} + \frac{1}{6}\operatorname{ad}_{\phi_b}^2 - \ldots\right) & 0 \\ 0 & I \end{bmatrix} \in \mathbb{R}^{(6+n)\times(6+n)}$$

with $\operatorname{ad}_p = \begin{bmatrix} \hat{p}_{4\ldots6} & \hat{p}_{1\ldots3} \\ 0 & \hat{p}_{4\ldots6} \end{bmatrix} \in \mathbb{R}^{6\times6}$ for $p \in \mathbb{R}^6$. This shows that the choice of coordinates (Q, v) has the required form.

From expression (3.3) for the twist of each link in the mechanism, we can derive an expression for the total kinetic energy. Let $I_i \in \mathbb{R}^{6\times6}$ denote the constant positive-definite diagonal inertia tensor of link i expressed in Ψ_i. The kinetic energy $U_{k,i}$ of link i then follows as

$$\begin{aligned} U_{k,i} &= \frac{1}{2}\left(V_{0i}^i\right)^T I_i V_{0i}^i \\ &= \frac{1}{2}\left(V_{0b}^b + J_i(q)\dot{q}\right)^T \operatorname{Ad}_{g_{ib}}^T I_i \operatorname{Ad}_{g_{ib}} \left(V_{0b}^b + J_i(q)\dot{q}\right) \\ &= \frac{1}{2}\begin{bmatrix} \left(V_{0b}^b\right)^T & \dot{q}^T \end{bmatrix} M_i(q) \begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix} = \frac{1}{2} v^T M_i(q) v \end{aligned} \tag{3.8}$$

with

$$M_i(q) := \begin{bmatrix} \operatorname{Ad}_{g_{ib}}^T I_i \operatorname{Ad}_{g_{ib}} & \operatorname{Ad}_{g_{ib}}^T I_i \operatorname{Ad}_{g_{ib}} J_i \\ J_i^T \operatorname{Ad}_{g_{ib}}^T I_i \operatorname{Ad}_{g_{ib}} & J_i^T \operatorname{Ad}_{g_{ib}}^T I_i \operatorname{Ad}_{g_{ib}} J_i \end{bmatrix} \tag{3.9}$$

The total kinetic energy of the mechanism is given by the sum of the kinetic energies of the mechanism links and the non-inertial base, that is,

$$U_k(q, v) = \frac{1}{2} v^T \underbrace{\left(\begin{bmatrix} I_b & 0 \\ 0 & 0 \end{bmatrix} + \sum_{i=1}^{n} M_i(q)\right)}_{\text{inertia matrix } M(q)} v \tag{3.10}$$

with $M(q)$ the inertia matrix of the total system. Note that neither $U_k(q, v)$ nor $M(q)$ depend on the pose g_{0b} and hence the choice of inertial reference frame Ψ_0.

We can write (3.4) in block-form as follows

$$\begin{bmatrix} M_{VV} & M_{qV}^T \\ M_{qV} & M_{qq} \end{bmatrix} \begin{bmatrix} \dot{V}_{0b}^b \\ \ddot{q} \end{bmatrix} + \begin{bmatrix} \bar{C}_{VV} & \bar{C}_{Vq} \\ \bar{C}_{qV} & \bar{C}_{qq} \end{bmatrix} \begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix} = \begin{bmatrix} F_b^b \\ \tau \end{bmatrix} \tag{3.11}$$

with F_b^b the external wrench on the base link, expressed in coordinates Ψ_b (such that it is collocated with the twist V_{0b}^b). To compute the matrix $\bar{C}(Q, v)$ for our system, we can use the observations that $M(q)$ is independent of g_{0b}, that $S(Q, \phi)$ is independent of q, and that $S(Q, 0) \equiv I$. Furthermore, the partial derivative of M with respect to ϕ_b is zero since M is independent of g_{0b}, and the second term of (3.5) is only non-zero for the $\bar{C}_{VV}$ block of $\bar{C}(Q, v)$.

The precise computational details of the partial derivatives follow the same steps as in the classical approach (Murray et al., 1994). To compute the partial derivatives of the adjoint matrices, one can use a relatively simple relation. If we express the velocity of joint k as $V_{(k-1)k}^{(k-1)} = X_k \dot{q}_k$ for constant X_k, then the following holds:

$$\frac{\partial \operatorname{Ad}_{g_{ij}}}{\partial q_k} = \begin{cases} \operatorname{Ad}_{g_{i(k-1)}} \operatorname{ad}_{X_k} \operatorname{Ad}_{g_{(k-1)j}} & \text{for } i < k \leq j \\ -\operatorname{Ad}_{g_{i(k-1)}} \operatorname{ad}_{X_k} \operatorname{Ad}_{g_{(k-1)j}} & \text{for } j < k \leq i \\ 0 & \text{otherwise} \end{cases}$$

To prove this, we start by writing out the spatial velocity of frame Ψ_k with respect to $\Psi_{(k-1)}$ when $i < k \leq j$:

$$\hat{X}_k \dot{q}_k = \hat{V}_{(k-1)k}^{(k-1)} = \dot{g}_{(k-1)k} g_{(k-1)k}^{-1} = \frac{\partial g_{(k-1)k}}{\partial q_k} g_{k(k-1)} \dot{q}_k$$

where $\hat{X} := \left[\begin{smallmatrix} \hat{X}_\omega & X_v \\ 0 & 0 \end{smallmatrix} \right]$. If we compare the first and the last terms, we get

$$\frac{\partial R_{(k-1)k}}{\partial q_k} = \hat{X}_\omega R_{(k-1)k}, \tag{3.12}$$

$$\frac{\partial p_{(k-1)k}}{\partial q_k} = \hat{X}_\omega p_{(k-1)k} + X_v. \tag{3.13}$$

We can use this relation in the expression for the partial derivative of $\operatorname{Ad}_{g_{(k-1)k}}$:

$$\begin{aligned} \frac{\partial \operatorname{Ad}_{g_{(k-1)k}}}{\partial q} &= \begin{bmatrix} \frac{\partial R_{(k-1)k}}{\partial q_k} & \frac{\hat{p}_{(k-1)k}}{\partial q_k} R_{(k-1)k} + \hat{p}_{(k-1)k} \frac{\partial R_{(k-1)k}}{\partial q_k} \\ 0 & \frac{\partial R_{(k-1)k}}{\partial q_k} \end{bmatrix} \\ &= \begin{bmatrix} \hat{X}_\omega & \hat{X}_v \\ 0 & \hat{X}_\omega \end{bmatrix} \begin{bmatrix} R_{(k-1)k} & \hat{p}_{(k-1)k} R_{(k-1)k} \\ 0 & R_{(k-1)k} \end{bmatrix} \\ &= \operatorname{ad}_{X_k} \operatorname{Ad}_{g_{(k-1)k}} \end{aligned} \tag{3.14}$$

It is now straight forward to show that

$$\begin{aligned} \frac{\partial \operatorname{Ad}_{g_{ij}}}{\partial q_k} &= \operatorname{Ad}_{g_{i(k-1)}} \frac{\partial \operatorname{Ad}_{g_{(k-1)k}}}{\partial q_k} \operatorname{Ad}_{g_{kj}} \\ &= \operatorname{Ad}_{g_{i(k-1)}} \operatorname{ad}_{X_k} \operatorname{Ad}_{g_{(k-1)k}} \operatorname{Ad}_{g_{kj}} \\ &= \operatorname{Ad}_{g_{i(k-1)}} \operatorname{ad}_{X_k} \operatorname{Ad}_{g_{(k-1)j}} . \end{aligned} \tag{3.15}$$

Similarly when $j < k \leq i$.

3.3.3 Manipulator Dynamics on a Forced Non-Inertial Base

We now simplify and specialize the dynamics equations by assuming that the motion of the platform is fully determined by external forces that are neither known nor of interest. We only assume that the relative pose g_{0b}, velocity V_{0b}^b, and acceleration $\dot{V}_{0b}^b$ of the base relative to the inertial world are known from measurements. This implicitly implies that the torques applied to the internal robot joints do not influence the motion of the platform, which is a reasonable assumption in our application of a relatively small robot attached to a large moving base.

Since we are not interested in the external forces on the mechanism, we can consider just the second block-row of (3.11), which expresses the robot accelerations $\ddot{q}$ as a function of the joint torques τ as well as the non-inertial motion of the base. This can be rewritten as

$$M_{qq}\ddot{q} + \bar{C}_{qq}\dot{q} + \underbrace{M_{qV}\dot{V}_{0b}^b + \bar{C}_{qV}V_{0b}^b}_{\text{inertial forces}} = \tau \tag{3.16}$$

which partially separates the usual robot dynamics (first two terms) from the inertial forces (third and fourth term), although the matrix $\bar{C}_{qq}$ generally still depends on V_{0b}^b. For a static base frame ($V_{0b}^b \equiv 0$), the equations reduce to the regular dynamics of an n-link robotic mechanism. Note that for constant V_{0b}^b, the terms due to the non-inertial base motion generally do not drop out, since a constant twist can also contain (non-inertial) angular components. Note also that neither the inertia of the base nor the second term in (3.5) appear in these equations.

The terms $\bar{C}_{qV}$ and $\bar{C}_{qq}$ can be written more explicitly as

$$\bar{C}_{qV} = \sum_{k=1}^{n} \frac{\partial M_{qV}}{\partial q^k}\dot{q}^k - \frac{1}{2}\frac{\partial^T}{\partial q}\left(\begin{bmatrix} M_{VV} & M_{qV}^T \end{bmatrix}\begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix}\right)$$

$$\bar{C}_{qq} = \sum_{k=1}^{n} \frac{\partial M_{qq}}{\partial q^k}\dot{q}^k - \frac{1}{2}\frac{\partial^T}{\partial q}\left(\begin{bmatrix} M_{qV} & M_{qq}^T \end{bmatrix}\begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix}\right)$$

This approach can be used to obtain the dynamics equations for an arbitrary n-link mechanism attached to a non-inertial base. Specific examples are presented in Section 3.5.

3.3.4 Gravitational Forces

Finally we include the gravitational forces. Let the wrench associated with the gravitational force of link i with respect to coordinate frame Ψ_i be given by

$$F_g^i = \begin{bmatrix} f_g \\ \hat{r}_g^i f_g \end{bmatrix} = -m_i g \begin{bmatrix} R_{0i}e_z \\ \hat{r}_g^i R_{0i}e_z \end{bmatrix} \tag{3.17}$$

where $e_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\mathsf{T}$ and r_g^i is the center of mass of link i expressed in frame Ψ_i. In our case Ψ_i is chosen so that r_g^i is in the origin of Ψ_i so we have $r_g^i = 0$. The equivalent joint torque associated with link i is given by

$$\tau_g^i = J_i(q)\,\mathrm{Ad}_{g_{0i}}^\mathsf{T}(Q)F_g^i(Q) \tag{3.18}$$

where J_i is the geometric Jacobian and $\mathrm{Ad}_{g_{0i}} = \mathrm{Ad}_{g_{0b}} \mathrm{Ad}_{g_{bi}}$ is the transformation from the inertial frame to frame i. We note that both R_{0i} and $\mathrm{Ad}_{g_{0i}}$ depend on the base configuration with respect to the inertial frame. The total effect of the gravity from all the links is then given by $\tau_g = \sum_{i=1}^{n} \tau_g^i$ which enters Equation (3.16) the same way as the control torque.

3.4 Compensation Using Motion Planning

In general there are two ways to deal with the inertial forces. We can try to compensate for the effects in the controller or in the motion planning algorithm. In the first method we cancel the effects of the inertial forces in the feed-forward terms of the controller. Consider the control law

$$\tau = \tau_{ff} + \tau_{PD} \tag{3.19}$$

where

$$\tau_{ff} = \underbrace{M_{qq}\ddot{q}_d + \bar{C}_{qq}\dot{q}_d}_{\text{tracking terms}} + \underbrace{M_{qV}\dot{V}_{0b}^b + \bar{C}_{qV}V_{0b}^b}_{\text{compensation for inertial forces}} - \underbrace{\sum_{n=1}^{n} (J_i \, \mathrm{Ad}_{g_{0i}}^{\mathsf{T}} \, F_g^i)}_{\text{gravity compensation}} \tag{3.20}$$

$$\tau_{PD} = \underbrace{K_P(q_d - q) + K_D(\dot{q}_d - \dot{q})}_{\text{PD-controller}} \tag{3.21}$$

This is the standard augmented PD control law which in our case also compensates for the inertial forces. As we are mainly interested in the feed-forward terms, we will assume perfect tracking, i.e. $q(t) = q_d(t)$. With this control law the non-inertial and gravitational terms are regarded as disturbances and are canceled.

When large inertial forces are present, canceling these terms may be very energy demanding. Thus, instead of regarding these terms as disturbances, we will find the trajectory for which the non-inertial and gravitational terms coincide with the tracking terms to an as large extent as possible. In doing this, the inertial forces will contribute to the desired motion instead of working against it. This will reduce the wear and tear on the manipulator, require less actuator torques and allow more accurate manipulation.

Given the dynamic equations, the initial position q_0, and desired end position q_{des} in joint coordinates, we want to find the optimal trajectory given by the minimum of the cost function P, i.e.

$$P_{min} = \min_{q(t)} \int_{t=T_0}^{T_1} P(\tau)\, dt \tag{3.22}$$

where $P(\tau)$ is some cost function representing the torque required for the motion,

$$\begin{aligned} q(T_0) &= q_0, \\ q(T_1) &= q_{des}, \end{aligned} \tag{3.23}$$

are the vectors describing the initial and end positions of all the joints and

$$M_{qq}\ddot{q} + \bar{C}_{qq}\dot{q} + M_{qV}\dot{V}_{0b}^b + \bar{C}_{qV}V_{0b}^b - \tau_g = \tau \tag{3.24}$$

determines the dynamics of the system.

The global solution to this problem is generally very complex. Assuming g_{0b}, $V_{0b}^b(t)$ and $\dot{V}_{0b}^b(t)$ known, we first need to compute $M_{qq}(q)$, $\bar{C}_{qq}(q, \dot{q}, V_{0b}^b)$, $M_{qV}(q)$ and $\bar{C}_{qV}(q, \dot{q}, V_{0b}^b)$. Then we need to find the optimal trajectory ($q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$) which requires the least amount of torque. Both these operations are computationally very demanding.

3.5 Examples

We now present specific examples of how the previous modeling and planning methods can be applied in case of specific robot motion objectives and given base motion. Here, we make specific choices as to how to discretize and approximate the problem to make it solvable; future work will investigate different and more general approaches.

3.5.1 Parameterization of Joint Motion

To reduce the search space, we assume that the shape of each joint trajectory is given so that we only need to find the starting time and the length of the motion for each joint. We also consider a cost function $P(\tau)$ that is quadratic in τ and thus reduce the problem to

$$P_{min} = \min_{t_0, t_1} \int_{t=T_0}^{T_1} \tau^\mathsf{T} D \tau \, dt \tag{3.25}$$

where D is a positive definite matrix that defines a metric in τ-space, $t_0 = \begin{bmatrix} t_{1,0} \cdots t_{n,0} \end{bmatrix}^\mathsf{T}$ are the starting times and $t_1 = \begin{bmatrix} t_{1,1} \cdots t_{n,1} \end{bmatrix}^\mathsf{T}$ are the end times for the n joints, which can all be chosen independently, with the restriction that $T_0 \leq t_{i,0} < t_{i,1} \leq T_1$ for all i and for a fixed prescribed time interval (T_0, T_1). We choose sinusoidal joint motions given by

$$\begin{aligned} \ddot{q}_i(t) &= a_i \sin\left(b_i(t - t_{i,0})\right), \\ \dot{q}_i(t) &= \frac{a_i}{b_i} - \frac{a_i}{b_i} \cos\left(b_i(t - t_{i,0})\right), \\ q_i(t) &= q_{i,0} + \frac{a_i}{b_i}(t - t_{i,0}) - \frac{a_i}{b_i^2} \sin\left(b_i(t - t_{i,0})\right), \end{aligned} \tag{3.26}$$

for $t \in (t_{i,0}, t_{i,1})$ and $q_i(t)$ constant otherwise. The boundary conditions $q_i(t_{i,0}) = q_{i,0}$ and $q_i(t_{i,1}) = q_{i,des}$ give rise to the following two equations

$$a_i = \frac{(q_{i,des} - q_{i,0}) b_i^2}{2\pi} \qquad\qquad b_i = \frac{2\pi}{t_{i,1} - t_{i,0}} \tag{3.27}$$

and hence the motion is fully parameterized by $t_{i,0}$ and $t_{i,1}$ for given q_0 and q_{des}. The motion planning problem is thus reduced to finding the optimal time intervals $(t_{i,0}, t_{i,1})$ for all joints $i = 1, \ldots, n$.

3.5.2 Base Notion

The environmental disturbances that affect the platform motion are wind, waves and ocean currents. The ocean currents are low frequency disturbances and will not affect the manipulator dynamics. It is common to assume the principle of superposition when considering wave and wind disturbances (Fossen, 2002) and they are normally modeled in the frequency spectrum. Many good models of the ship motion for different sea states are available in literature (Fossen, 2002; Salvesen et al., 1970).

The platform motion is modeled as $g_{0b}(t) \in SE(3)$. Large marine vessels are often found to have a characteristic motion which we can represent as a vector subspace of $se(3)$. For the purpose of this paper, we will estimate the main angular motion of the platform somewhat roughly by a sinusoidal motion in $SO(2)$. Assume that the waves hit the platform with a velocity in the direction of the y-axis in Ψ_b. The platform pose and acceleration are then given by an angular motion about the x-axis:

$$\phi = A\sin(Bt) \tag{3.28}$$

This is a simplified motion, but the dynamic equations are valid for any motion in $SE(3)$. Specific examples of how this base motion affects the manipulator motion are shown in the following.

3.5.3 1-DoF Manipulator

Consider first a mechanism with a single 1-DoF prismatic joint located at $p_{b1} = \begin{bmatrix} 0 & 0 & l_1 \end{bmatrix}^\mathsf{T}$ in Ψ_b and moving in the direction of the y-axis. Let the base motion be given as in (3.28). We set $m = 1$ and the dynamics equations reduce to

$$\tau = \ddot{q} - l_1\ddot{\phi} - q\dot{\phi}^2 - g\sin(\phi). \tag{3.29}$$

We start by approximating the base motion given in (3.28) by the Taylor approximation

$$\sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \mathcal{O}(x^9). \tag{3.30}$$

We can write $\phi(t)$, $\dot{\phi}(t)$ and $\ddot{\phi}(t)$ as

$$\begin{aligned}
\phi(t) &\approx A\left(Bt - \frac{(Bt)^3}{3!} + \frac{(Bt)^5}{5!} - \frac{(Bt)^7}{7!} + \mathcal{O}(t^9)\right),\\
\dot{\phi}(t) &\approx AB\left(1 - \frac{(Bt)^2}{2!} + \frac{(Bt)^4}{4!} - \frac{(Bt)^6}{6!} + \mathcal{O}(t^8)\right),\\
\ddot{\phi}(t) &\approx AB^2\left(-\frac{(Bt)}{1!} + \frac{(Bt)^3}{3!} - \frac{(Bt)^5}{5!} + \mathcal{O}(t^7)\right).
\end{aligned}$$

This is typically a good approximation for one period of sinusoidal motion. We approximate the desired joint motion given by (3.26) in the same way.

As we have only one joint we set $t_0 = T_0$ and $t_1 = T_1$. The minimization problem is then reduced to

$$P_{min} = \min_{t_0,t_1} \int_{t_0}^{t_1} (\ddot{q} - l_1\ddot{\phi} - q\dot{\phi}^2 - g\sin(\phi))^2 dt \tag{3.31}$$

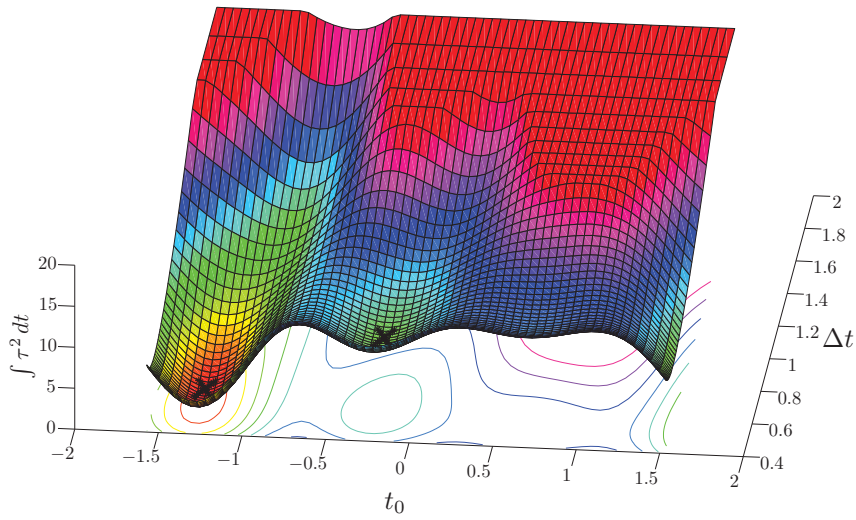


Figure 3.2: The torques needed to move the prismatic joint from $q_0 = 0$ to $q_{des} = 1.5$ for high frequency base motion $\phi(t) = -1/5 \sin(2t)$ plotted with respect to the start time t_0 and the motion length $\Delta t = t_1 - t_0$. The minima found are marked with an "X".

which after substituting the Taylor approximations reduces to the problem of finding the minimum of a polynomial equation. We can now quickly find the optimal solution with respect to the start and end times t_0 and t_1. We define the search space as the time interval for which the Taylor approximation is accurate, i.e., $(t_0, t_1) \in (-t_T, t_T)$ where $2t_T$ is the wave period of the principal frequency of the waves. Fig. 3.2 illustrates the value of the integral (3.31) for different start and end times for $\phi(t) = -1/5 \sin(2t)$, $q_0 = 0$ and $q_{des} = 1.5$. The optimal and worst case trajectories are shown in Fig. 3.3.

3.5.4 4-DoF Manipulator

The previous example can be solved efficiently as it reduces to finding the minimum of a polynomial equation. As a second example, we show how numerical methods can be used to compute optimal motion paths for the 4-DoF manipulator shown in Fig. 3.1 with realistic mass and inertia parameters. We now use the exact equations for base motion and the manipulator dynamics. The base moves along the angular motion pattern (3.28) at a relatively low frequency, which means the inertial forces mostly enter through a changing direction of gravity.

We solve the motion planning problem by numerically minimizing the objective function (3.22) and parameterize the problem as follows: each joint trajectory is given by a separate sinusoidal motion (3.26) with parameters $t_{i,0}$ and $t_{i,1}$, the total motion from start to goal is to be finished within a fixed prescribed time interval $(T_0, T_1) = (0, 10)$, and the cost function is chosen as (3.25) with $D = 10^{-6} I$ and integrated over the fixed time

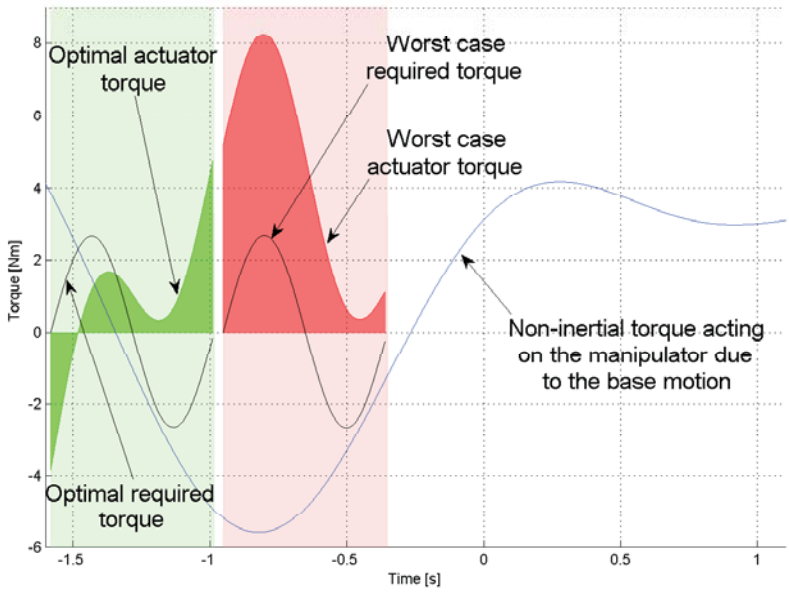


Figure 3.3: Optimal and worst case trajectories. The required torque is the total torque required for the desired joint motion, which is the sum of the actuator and the inertial torques. The actuator torque is the torque applied by the actuator so that the total torque is equal to the required torque, i.e. the inertial torques subtracted from the required torque. This is the torque to be minimized. The optimal interval is found at $t = [-1.58\ -0.98]$. The worst case is found at $t = [-0.95\ -0.35]$. The integrated torque (squared) for the optimal solution is 0.87 and 11.38 for the worst case. As the length of the motion increases (Δt increases in Fig. 3.2), the integrated torque increases unbounded, thus the worst case starting point search is performed with a fixed motion length of 0.6s.

interval (T_0, T_1). We choose the start and end configurations as

$$q(0) = \begin{bmatrix} -2.5 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$$
$$q(T) = \begin{bmatrix} 2.5 & \frac{\pi}{2} & \frac{\pi}{2} & \frac{\pi}{2} \end{bmatrix}^\mathsf{T}$$

and the base motion as (3.28) with $A = \frac{5}{4\pi}$ and $B = \frac{2\pi}{5}$. The motion planning problem thus amounts to finding the eight parameters (one start and end time for each joint) that minimize the total squared required torque integrated over a fixed time interval while starting and finishing the robot in the required configurations.

Figures 3.4 and 3.5 show the solution obtained using Matlab's constrained minimization function fmincon.mat. A full animation of the resulting motions can be found in the video accompaniment to this paper. Fig. 3.4 shows still shots of the optimal solution and illustrates the sinusoidal motion of the base. Fig. 3.5 compares three solutions: one baseline solution that simply takes the start time for each joint trajectory at $t_{i,0} = T_0$ and the finish time at $t_{i,1} = T_1$, one solution that optimizes the cost function assuming zero base motion

(a horizontal stationary base), and one solution that optimizes the cost function taking the real base motion into account. The associated costs are 19, 21, and 11, respectively.

The figure shows that, for this example, taking the base motion into account can significantly reduce the cost and hence the required torque. The joint motions optimized for a static base (dashed line) even perform worse than the baseline joint motions (dotted line) when applied during non-static base motion. When optimizing the joint motions while taking the base motion into account (solid line), the result is much improved, and the benefit of the resulting motions can be understood intuitively: the prismatic motion is delayed and shortened as to optimally use the changing gravity direction (due to base rotation) in the acceleration and deceleration phase, similar to the previous 1-DoF example. The resulting required actuator torque τ_1 is thus reduced to close to zero during the motion, i.e., in the time interval $(t_{1,0}, t_{1,1}) = (3.2, 7.2)$. Similarly, the motion of joints 3 and 4 is delayed as to minimize the amount of time spent holding up the links against gravity.

The example shows how knowledge of the base motion can be used to significantly reduce torque requirements, even with only little freedom in the optimization (only $t_{i,0}$ and $t_{i,1}$ can be optimized). If the shapes of the trajectories are allowed to be changed and optimized in more detail, improvement should be even more significant. Including more parameters makes the optimization problem more complex, though, and numerical solutions may get more easily trapped in local minima.

3.6 Conclusions

The classical dynamics equations for a serial manipulator have been extended to also include the motion of a forced non-inertial base. The dynamics equations are derived using a generalized Lagrangian method. This allows us to model the base motion as a "free motion" joint serially connected with the 1-DoF joints of the manipulator.

Examples for a 1-DoF and 4-DoF manipulator mounted on a platform are presented. We include the platform motion in the dynamics and find the trajectory that takes the manipulator from an initial position to an end position with the least amount of torque and compare this with the optimal trajectory when the platform is assumed not to be moving. The simulations show that when the ship motion is known the amount of torque needed for a given task can be substantially reduced if the inertial forces are taken into account.

A possible extension for future work is to optimize the shape of the joint trajectories with more variables. Adding more details to the joint trajectories should increase performance even more. If a sufficiently accurate model of the platform can be obtained, this may allow us to compensate for the inertial forces in high accuracy applications. Another interesting topic for future work is to look into how model predictive control can be used to compensate for the platform motion.

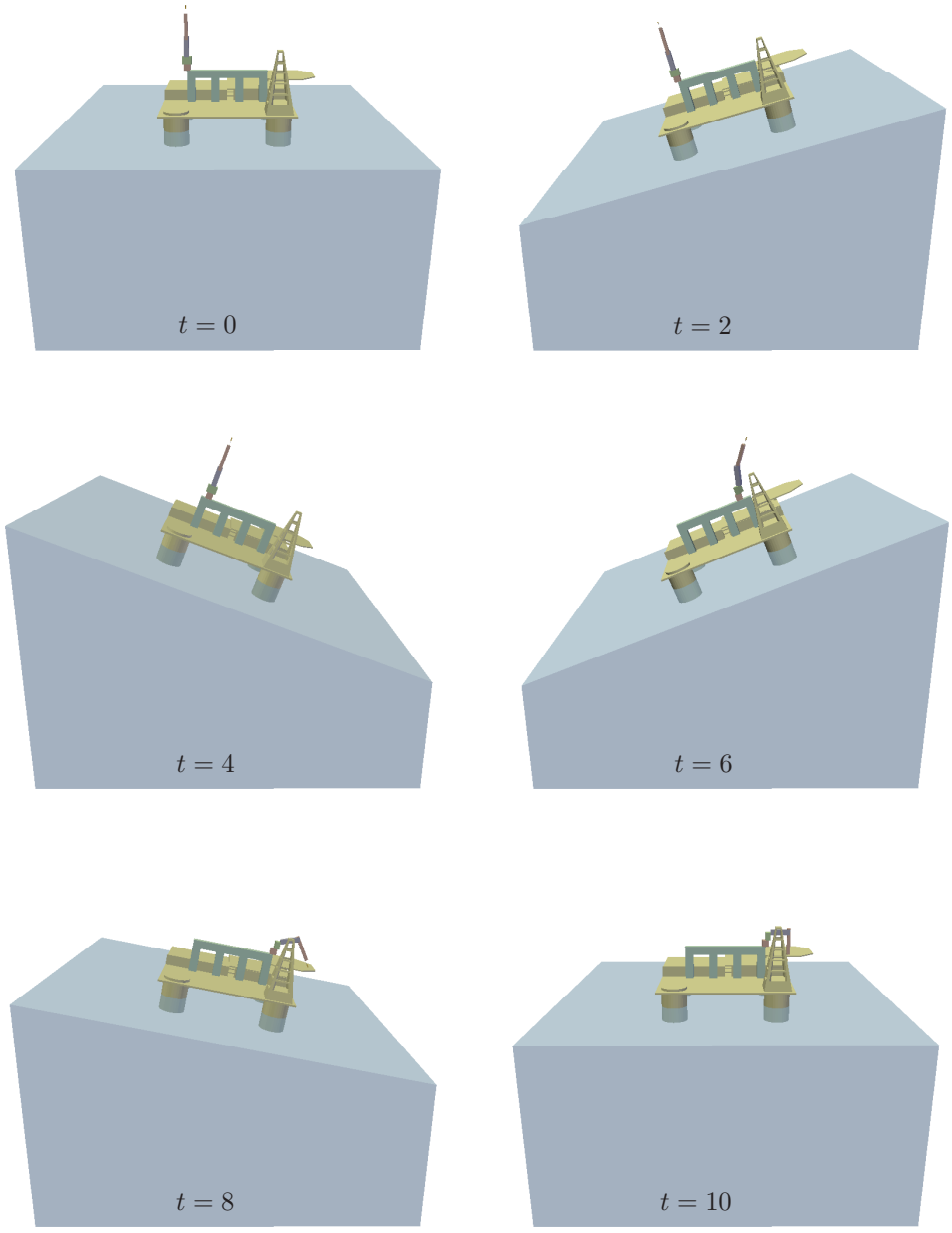
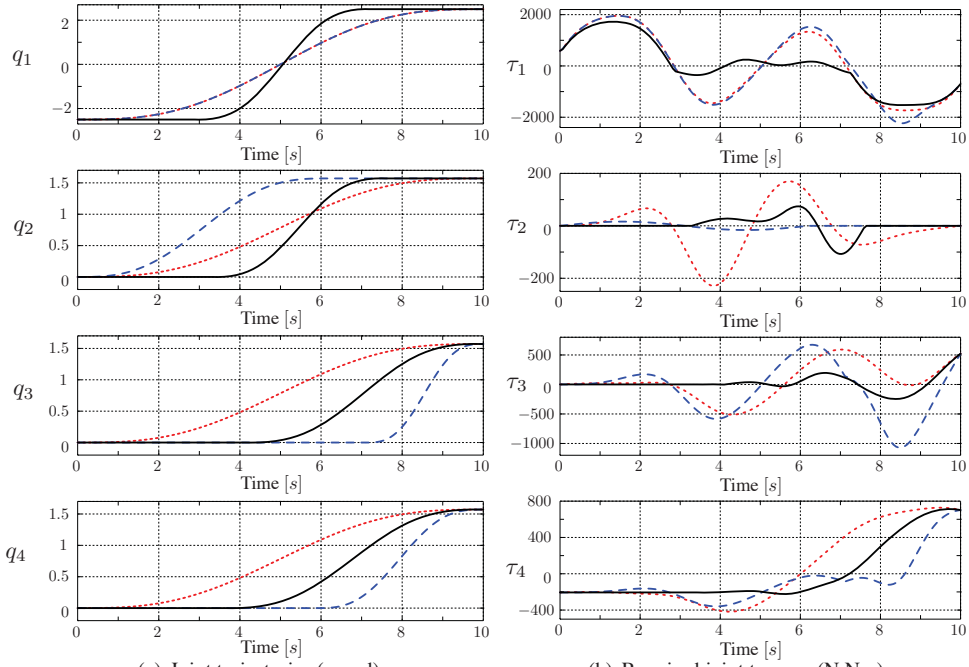


Figure 3.4: Still shots from the simulation of the fully optimized trajectory corresponding to the solid lines in Fig. 3.5.



(a) Joint trajectories (m,rad). (b) Required joint torques (N,Nm).

Figure 3.5: Optimal motion trajectory for a 4-DoF manipulator. Three different trajectories are shown: a baseline trajectory with maximum motion duration (dotted lines), an optimized trajectory assuming zero base motion (dashed lines), and an optimized trajectory taking the correct non-zero base motion into account (solid lines).

Chapter 4

Motion Planning and Control of Robotic Manipulators on Seaborne Platforms

4.1 Abstract

Robots on ships have to endure large inertial forces due to the non-inertial motion of the ship. It is thus important to investigate to what extent it is possible to predict the future motion of a ship. The ship motion affects both the motion planning and control of the manipulator and accurate predictions can improve performance substantially. Based on these predictions, this paper presents a new approach to motion planning and control of such manipulators. It is shown that the effects of the non-inertial forces can be eliminated—in fact, the robot can even leverage the inertial forces to improve performance compared to robots on a fixed base. To perform realistic experiments a 9-DoF robot is used. The first five joints are used to generate the real ship motion, and the last four joints are used for motion planning. The dynamic coupling between the first five and the last four joints is thus exactly the same as the dynamic coupling between a ship and a manipulator, which allows for very realistic experiments.

Keywords: Marine Systems, Robotics, Stochastic systems, MPC.

4.2 Introduction

Robotic manipulators on non-inertial platforms such as ships have to endure large inertial forces due to the non-inertial motion of the platform. When the non-inertial platform's motion is known, motion planning and control algorithms can try to eliminate these perturbations. In some situations the motion planning algorithms can even leverage the inertial forces to more economically move to a target point. However, for many non-inertial platforms, the motion is unknown.

This paper first investigates to what extent it is possible to predict the future motion of a ship. Using real ship motion measurements one can study how the uncertainty of the prediction algorithms changes with the prediction horizon. Then a new motion planning approach that, based on the predicted future motion of the ship, finds the optimal trajectory from an initial to a target configuration is presented. It is also shown that by including the uncertainty in the cost function the maximum torques needed to reach the target configuration can be reduced.

Due to the stochastic nature of the ship motion and the dynamic coupling between the ship and the manipulator, empirical studies are extremely important to validate both the ship motion predictions and the motion planning algorithms. To perform realistic experiments a 9-DoF robot is used. The first five joints are used to generate the real ship motion, and the last four joints are used for motion planning. The dynamic coupling between the first five and the last four joints is thus exactly the same as the dynamic coupling between a ship and a manipulator. It is thus possible to perform very realistic experiments as ship motions measured from a real ship are used to generate the actual motion and at the same time realistic ship motion predictions are used as inputs to the motion planner. To get statistically meaningful results several simulations are performed to confirm the experimental results.

Ships and other seaborne platforms are expected to become increasingly unmanned in the future and hence the need for autonomously operating robots for surveillance, maintenance, and operation will continue to increase over time (Love et al., 2004; Kitarovic et al., 2005). The demand for unmanned operation becomes even higher in harsh environments such as high sea state (Figure 4.1), when it can be dangerous for human operators to be exposed. High sea environments are not only dangerous to human operators, they also pose significant challenges for robotic control: Large inertial forces will influence the manipulator and, when not anticipated and accounted for, can make the operation inaccurate, extremely energy demanding, and sometimes even impossible due to torque limits. The inertial forces thus need to be taken into account in the motion planning of the robot.

In From et al. (2009a) the authors solve the problem of optimal motion planning for a robot mounted on a ship *under the assumption the base motion is known for all times*. The approach includes the ship motion in the trajectory planning problem and an optimal trajectory in terms of actuator torques is found. However, in most practical situations the forces acting on the ship due to the interaction with waves and wind are very irregular and one cannot expect to know the base motion for all times.

The extent to which it is possible to obtain accurate ship motion predictions can thus directly affect how well the inertial forces can be compensated for or even taken advantage of. However, the accuracy of the ship motion prediction not only directly determines how optimal a solution one can achieve, it also affects the computational requirements. In a receding horizon setting, where the optimal control input sequence is re-computed at regular intervals, the computational burden will increase for an inaccurate model: for an inaccurate model the initialization point taken from the previous solution is further away from the optimal solution. In addition to affecting the choice of horizon the modeling error thus directly determines the frequency for which the optimal control or optimal trajectory can be recalculated.

An important contribution in this paper is the use of real empirical data, both for the experiments and the simulations. Much of the literature on ship motion prediction uses

Figure 4.1: A ship in high sea. The waves forces can result in very high accelerations in the ship motion.

computer generated data to verify the accuracy of the prediction algorithms, which leads to unrealistically small errors in the predictions. An important difference of the work presented here is thus the use of real full-scale ship motion data to test the performance of the ship motion prediction algorithms.

Another important contribution is the realistic experiments. For the experiments a 9-DoF robot was used: The real ship motion was fed into the first five joints, generating a very realistic ship motion, and the last four joints are then used for optimal motion planning for a 4-DoF robot on a moving base. This allows for experiments performed on a 4-DoF robot mounted on a base with exactly the same motion as if the robot had been mounted on a ship. The predictions used are based on the real ship motion and the experiments thus give important information on to what extent the inertial forces can be compensated for.

Experiments are important to study how the motion of the base affects the manipulator dynamics. The experiments also allow for direct measurements of the torques that act on the manipulator due to the inertial forces. This paper thus presents, for the first time, a detailed study of how the inertial motion of the ship maps to the joint torques. This is used to show that the forces that act on the manipulator due to the waves are excessive and cannot be ignored in the motion planning and control of the manipulator.

Stochastic uncertainty is present in a wide variety of systems, ranging from mechanical systems and process control to finance. In general, receding horizon control is a well suited control scheme to deal with uncertainties, but most approaches do not use information about the probability distribution governing the uncertainty and they only assume that the uncertainty is bounded. Thus, the information about the probabilistic distribution is ignored and the worst-case representation of the disturbances or constraints often leads to a conservative solution.

In this paper a new motion planning algorithm that also minimizes the variance of the

controlled state is presented. First, real ship motion measurements are used to calculate the variance of the predictions of the forced state. It is found that the variance is different for roll, pitch and yaw. Second, it is shown how to use the Extended Kalman filter to find how the variance in the forced state maps to the variance of the controlled state, i.e., how the uncertainties in the ship motion predictions map to uncertainties in the robot state. The general idea of this approach is to exploit the fact that there are some components of the ship motion that are more difficult to predict than others. Also, for different configurations of the robot, the inertial forces will affect the robot differently. Thus, by including the variance in the cost function one can force the motion planner to choose a trajectory that is less affected by the largest and most uncertain components of the base motion. When a receding horizon approach is applied, it is found that by augmenting the cost function to also include the variance it is possible to choose a longer horizon than when the variance is not included.

The paper is organized as follows: Section 4.3 gives a short introduction to ship-manipulator modeling, presented in more detail in From et al. (2009a). The ship motion prediction algorithms used are presented in Section 4.4 and in Section 4.5 it is shown how to use these predictions to improve the motion planning and control of a robotic manipulator on a moving base. The simulations and empirical studies are presented in Section 4.6. Related research and references are discussed in Section 4.7.

4.3 Ship-Manipulator Modeling

In From et al. (2009a) the classical dynamic equations for a serial manipulator arm with 1-DoF joints were extended to include the forced 6-DoF motion of the base. For more details on how to derive the dynamics see From et al. (2009a) and Duindam and Stramigioli (2007, 2008). Consider the setup of Fig. 4.2 describing a general n-link robot manipulator arm attached to a moving base and choose an inertial coordinate frame Ψ_0, a frame Ψ_b rigidly attached to the moving base, and n frames Ψ_i (not shown) attached to each link i at the center of mass. Finally, choose a vector $q \in \mathbb{R}^n$ that describes the configuration of the n joints. Using standard notation (Murray et al., 1994), the pose of each frame Ψ_i relative to Ψ_0 can be described as a homogeneous transformation matrix $g_{0i} \in SE(3)$. This pose can also be described using the vector of joint coordinates q as

$$g_{0i} = g_{0b} g_{bi}(q). \tag{4.1}$$

The base pose g_{0b} and the joint positions q thus fully determine the configuration of the robot.

In a similar way, the spatial velocity of each link can be expressed using twists (Murray et al., 1994):

$$V_{0i}^0 = \begin{bmatrix} v_{0i}^0 \\ \omega_{0i}^0 \end{bmatrix} = V_{0b}^0 + V_{bi}^0 = \mathrm{Ad}_{g_{0b}} \left(V_{0b}^b + J_i(q)\dot{q} \right) \tag{4.2}$$

where v_{0i}^0 and ω_{0i}^0 are the linear and angular velocities, respectively, of link i relative to the inertial frame, $J_i(q) \in \mathbb{R}^{6\times n}$ is the geometric Jacobian of link i relative to Ψ_b, the adjoint is defined as $\mathrm{Ad}_g := \left[\begin{smallmatrix} R & \hat{p}R \\ 0 & R \end{smallmatrix}\right] \in \mathbb{R}^{6\times 6}$, and $\hat{p} \in \mathbb{R}^{3\times 3}$ is the skew symmetric matrix such

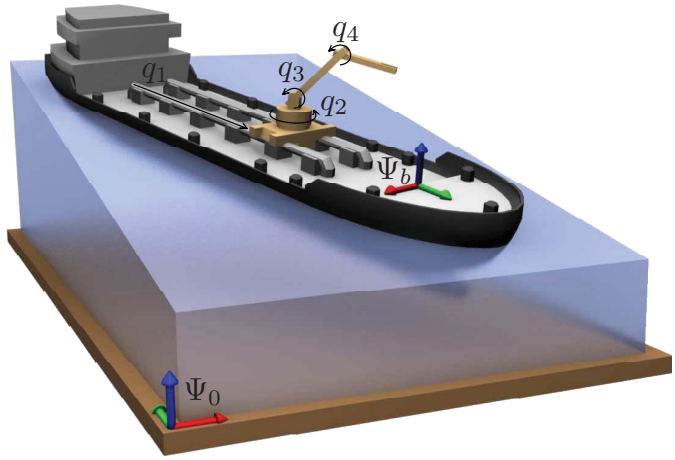


Figure 4.2: Model setup for a four-link robot attached to a non-inertial base with coordinate frame Ψ_b. Frame Ψ_0 denotes the inertial reference frame.

that $\hat{p}x = p \times x$ for all $p, x \in \mathbb{R}^3$. The velocity state is thus fully determined given the twist V_{0b}^b of the base and the joint velocities $\dot{q}$.

The dynamic equations can be written in block-form as follows

$$\begin{bmatrix} M_{VV} & M_{qV}^T \\ M_{qV} & M_{qq} \end{bmatrix} \begin{bmatrix} \dot{V}_{0b}^b \\ \ddot{q} \end{bmatrix} + \begin{bmatrix} C_{VV} & C_{Vq} \\ C_{qV} & C_{qq} \end{bmatrix} \begin{bmatrix} V_{0b}^b \\ \dot{q} \end{bmatrix} = \begin{bmatrix} F_b^b \\ \tau \end{bmatrix} \tag{4.3}$$

with F_b^b the external wrench on the base link, expressed in coordinates Ψ_b (such that it is collocated with the twist V_{0b}^b).

This paper is concerned with the effects that the base pose g_{0b}, the base velocity V_{0b}^b, and the base acceleration $\dot{V}_{0b}^b$ have on the manipulator dynamics. This can be seen by rewriting the dynamics as

$$M_{qq}\ddot{q} + C_{qq}\dot{q} + \underbrace{M_{qV}\dot{V}_{0b}^b + C_{qV}V_{0b}^b}_{\text{inertial forces}} = \tau. \tag{4.4}$$

Finally, the way the gravitational forces map to joint torques depends on the configuration of the base and is added to the right hand side of (4.4). The torque associated with link i is given by

$$\tau_g^i = J_i(q)\,\mathrm{Ad}_{g_{0i}}^{\mathsf{T}}(Q)F_g^i(Q). \tag{4.5}$$

Note that both R_{0i} and $\mathrm{Ad}_{g_{0i}}$ depend on the base configuration with respect to the inertial frame. F_g^i is given by

$$F_g^i = \begin{bmatrix} f_g \\ \hat{r}_g^i f_g \end{bmatrix} = -m_i g \begin{bmatrix} R_{0i}e_z \\ \hat{r}_g^i R_{0i}e_z \end{bmatrix} \tag{4.6}$$

where $e_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}}$ and r_g^i is the center of mass of link i expressed in frame Ψ_i.

4.4 Ship Motion Prediction

Due to the stochastic nature of the forces that act on ships, ship motion prediction is a very difficult problem. This section presents two simple and computationally efficient methods for predicting the future motion of the ship: the Auto-Regressive (AR) predictor and a predictor using a superposition of sinusoidal waves representation. In this paper the focus is on these simple and computationally efficient methods as they do not require a ship or wave model, nor external sensors such as wave cameras or force sensors. Adding sensors or using more advanced methods will allow for more accurate predictions and the results presented in the following sections can thus probably be improved if external sensors are added.

The Auto-Regressive (AR) predictor is an all-pole model (i.e., no inputs) and gives an estimate of the output directly without the need for information about the forces that cause the motion. Write

$$y(t) = -a_1 y(t-1) - a_2 y(t-2) - \cdots - a_n y(t-n) \tag{4.7}$$

and define

$$\begin{aligned} \phi(t) &= \begin{bmatrix} -y(t) & -y(t-1) & -y(t-2) & \cdots & -y(t-n+1) \end{bmatrix}^\mathsf{T}, \\ \theta &= \begin{bmatrix} a_1 & a_2 & \dots a_n \end{bmatrix}^\mathsf{T}. \end{aligned} \tag{4.8}$$

Collecting N samples and stacking ϕ in Φ and y in Y one can find the optimal parameters θ in the least squares sense by

$$\theta = (\Phi^\mathsf{T}\Phi)^{-1}\Phi^\mathsf{T} Y \tag{4.9}$$

and the prediction problem is solved by

$$y(t+1) = \phi(t)^\mathsf{T}\theta. \tag{4.10}$$

Alternatively one can fit the superposition of N sine waves to the measurements in the least squares sense. Following the approach in Chung et al. (1990) write

$$\xi(t) = \sum_{i=1}^{N} A_i \sin(\omega_i t + b_i) \tag{4.11}$$

where A_i is the amplitude of the sines, ω_i is the frequency and b_i is the phase. Assuming the frequencies are found from the peaks in the frequency spectrum, the problem amounts to finding A_i and b_i. Note that

$$\xi(t) = \sum_{i=1}^{N} a_{2i-1}\sin(\omega_i t) + a_{2i}\cos(\omega_i t) \tag{4.12}$$

where $a_{2i-1} = A_i\cos(b_i)$ and $a_{2i} = A_i\sin(b_i)$ are used to handle phase shifts. The parameters

$$\theta = \begin{bmatrix} a_1 & a_2 & \cdots & a_{2N} \end{bmatrix}^\mathsf{T} \tag{4.13}$$

representing the best fit in the least squares sense are then found from (4.9) with

$$\phi(t) = \begin{bmatrix} \sin(\omega_1 t) & \cos(\omega_1 t) & \cdots & \sin(\omega_N t) & \cos(\omega_N t) \end{bmatrix}^\mathsf{T}.$$

4.5 Motion Planning and Control in a Stochastic Environment

This section discusses the motion planning problem, i.e., to take the manipulator from an initial configuration to a target configuration using as little torque as possible. By planning the motion so that the inertial forces contribute to the motion and don't work against it, it is possible, in addition to save energy, to achieve more accurate trajectory tracking and reduce the strain and tension on the manipulator.

4.5.1 Motion Planning

Consider the control law

$$\tau = \tau_{ff} + \tau_{PD} \tag{4.14}$$

where

$$\tau_{ff} = \underbrace{M_{qq}\ddot{q}_d + C_{qq}\dot{q}_d}_{\text{tracking terms}} + \underbrace{M_{qV}\dot{V}_{0b}^b + C_{qV}V_{0b}^b}_{\text{compensation for inertial forces}} \underbrace{- \sum_{n=1}^{n} (J_i \operatorname{Ad}_{g_{0i}}^{\mathsf{T}} F_g^i)}_{\text{gravity compensation}} \tag{4.15}$$

$$\tau_{PD} = \underbrace{K_P(q_d - q) + K_D(\dot{q}_d - \dot{q})}_{\text{PD-controller}}. \tag{4.16}$$

This is the standard augmented PD control law which in this case also compensates for the inertial forces. Based on the predictions of g_{0b}, V_{0b}^b, and $\dot{V}_{0b}^b$ for a given horizon this control law tries to cancel these disturbances regardless of whether they contribute to the desired motion or not.

For trajectory tracking this is in general a very energy demanding solution. When large inertial forces are present, simply canceling these terms as in Equation (4.15) may require excessive joint torques. Thus, instead of regarding these terms as disturbances, the prediction of the ship motion can be included in the motion planning. The planner can then use this information to calculate the trajectory that requires the least actuator torque for the given base motion. As an example, consider a manipulator that is to move from the left to the right on the ship. If it chooses to start the trajectory at a time when the inertial forces contribute to the desired motion it can potentially get an almost free ride from one side to the other. If it simply chooses to cancel these disturbances, for example by (4.15), it might end up following a trajectory for which the inertial forces are working against the desired motion for the entire interval. One intuitive situation where this can occur is when the manipulator moves uphill instead of downhill for the entire motion and will thus not take advantage of the gravitational forces.

This paper follows the approach presented in From et al. (2009a) and solves the motion planning problem by numerically minimizing an objective function representing the joint torques with respect to the start and end time. The cost function is found from the whole interval (T_0, T_1) and not only the interval (t_0, t_1) when the motion occurs, i.e.,

$$P = \min_{t_0, t_1} \int_{t=T_0}^{T_1} \tau^{\mathsf{T}} D\tau \, dt \tag{4.17}$$

where $T_0 \leq t_0 < t_1 \leq T_1$ and D is a positive definite matrix that defines a metric in τ-space. In From et al. (2009a) this was solved assuming the base motion was known. This paper uses realistic predictions of the base motion in the cost function. These predictions will become less accurate as the horizon increases, and it is investigated how the choice of motion planning algorithm and prediction horizon affects the performance of the motion planner.

4.5.2 Stochastic MPC

As will be clear in Section 4.6.1 the accuracy of the predictions is different for the different axes. For example for the AR predictor the predictions of the angular acceleration about the y-axis is less accurate than the x- and z-axes. Also, the linear acceleration in the direction of the x-axis is far more accurate than the y- and z-axes. Thus, this section presents a modified cost function that minimizes also the expected variance on the output assuming information about how the variance evolves with time is available for the different components of the ship motion.

In Cannon et al. (2007) the control objective of the stochastic model predictive control (stochastic MPC) law is to regulate the expected value and variance of the output state. In this section the same ideas are applied and by including the covariance matrix in the cost function a trajectory that also minimizes the variance is chosen. The cost function as defined in Cannon et al. (2007) is given by

$$P = \sum_{j=0}^{N-1} l(k+j|k) + L(k+N|k) \tag{4.18}$$

where L is the cost-to-go function and

$$l(k+j|k) = \bar{x}^2(k+j|k) + \kappa^2 \sigma_x^2(k+j|k) \tag{4.19}$$

with

$$\bar{x}(k+j|k) = \mathbb{E}_k(x(k+j|k)) \tag{4.20}$$

$$\sigma_x^2(k+j|k) = \mathbb{E}_k\left[x(k+j|k) - \bar{x}(k+j|k)\right]^2 \tag{4.21}$$

denoting the expected value and variance of $x(k+j|k)$. The relative weighing of the expected value and the variance can thus be controlled directly through the parameter κ.

In the following the same ideas are applied and the covariance matrix is included in the cost function so that a trajectory that also minimizes the covariance is chosen. A cost function similar to the one found in Cannon et al. (2007) is given by

$$P = \sum_{j=0}^{N-1} l(k+j|k) + L(k+N|k) \tag{4.22}$$

where $L(k+N|k)$ is the cost-to-go and

$$l(k+j|k) = \bar{q}^2(k+j|k) + \kappa \left|\left|\Sigma_q(k+j|k)\right|\right|_r \tag{4.23}$$

with $\bar{q}(k+j|k) = \mathbb{E}[q(k+j|k)]$ denoting the expected state of the robot and

$$\Sigma_q = \begin{bmatrix} \mathbb{E}[(q_1 - \bar{q}_1)(q_1 - \bar{q}_1)] & \dots & \mathbb{E}[(q_1 - \bar{q}_1)(q_n - \bar{q}_n)] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[(q_n - \bar{q}_n)(q_1 - \bar{q}_1)] & \dots & \mathbb{E}[(q_n - \bar{q}_n)(q_n - \bar{q}_n)] \end{bmatrix} \tag{4.24}$$

so that $\Sigma_q(k+j|k)$ is the covariance matrix of $q(k+j|k)$ given the measurements of the robot state $q(i)$ and the ship state $g_{0b}(i)$ for $i = k_0 \dots k$, and $||\cdot||_r$ denotes the Euclidean norm of each row. Similarly, the second part of (4.19) can be added to the cost in (4.17) to get the cost function in the form

$$P = \min_{t_0, t_1} \int_{t=T_0}^{T_1} \left(\tau^{\mathsf{T}}(t) D \tau(t) + \kappa \, ||\Sigma_q(t|T_0)|| \right) dt. \tag{4.25}$$

where $||\Sigma_q||$ denotes the Euclidean norm of the covariance matrix. The problem is thus to find the start time $t_{i,0}$ and the end time $t_{i,1}$ for the motion of each joint subject to the restriction $T_0 \leq t_{i,0} < t_{i,1} \leq T_1$. The cost, however, sums over the entire pre-defined interval (T_0, T_1), i.e., also when the joints do not move and $\dot{q}_i(t) = 0$. Note that in this case the joint torques are not necessarily zero because of the inertial forces.

Assume that each degree of freedom of the ship motion has a normal distribution, i.e.,

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \vdots \\ \dot{r} \end{bmatrix} \sim \begin{bmatrix} \mathcal{N}(\bar{\dot{u}}, \sigma^2_{\dot{u}}) \\ \mathcal{N}(\bar{\dot{v}}, \sigma^2_{\dot{v}}) \\ \vdots \\ \mathcal{N}(\bar{\dot{r}}, \sigma^2_{\dot{r}}) \end{bmatrix} \tag{4.26}$$

Examples of the expected value and the variance for the acceleration of the ship are shown in Figures 4.4 and 4.5, respectively. Similar relations can also be found for the position and velocity of the ship.

For linear systems the Kalman filter can be used to find the expected state and the error covariance. For non-linear systems the extended Kalman filter is implemented, i.e., linearizing around the mean value, and find the expected state of the robot and the covariance matrix used in (4.25). It is then possible to include both the expected state and the covariance at time $(k+j)$ given the measurements available at time k also for non-linear systems. This is then included in the cost function (4.25) and the optimal solution is found by minimizing the weighed cost of the expected value and the covariance.

4.6 Simulations and Experimental Studies

This section presents the simulation results and the results from experiments performed in the lab. Due to the stochastic nature of the problem, both experimental and simulation studies are important. The experimental studies are carried out in order to gain insight into how the moving base and errors in the prediction of the base motion affect the motion planning and control of the manipulator. The experiments make it possible to measure the torques directly and get valuable insight into the complex coupling between the ship and manipulator and how this affects the motion planning and control of the manipulator. Also, due to the stochastic nature of the disturbances, simulations are important to be able to perform a sufficient number of runs and hence get statistically meaningful results.

Figure 4.3: The Emmy Dyvi class "Erik Bye" used to measure the ship motions. "Erik Bye" is owned by Redningsselskapet, Norway.

4.6.1 Empirical Data of the Ship Motion

The need for empirical data is of utmost importance when verifying the performance of prediction algorithms. Most publications on the topic of ship motion prediction use computer generated data such as a combination of sines, a wave model or a sine with added noise. This will not result a in good performance indicator because of the stochastic nature of the waves. In this work measurements from a real full-scale ship are used. This makes it possible to compare the different prediction algorithms on real data and, most importantly, it gives valuable information about the accuracy of the predictions for different prediction horizons. The ship used to collect the measurements was the RS 113 "Erik Bye", shown in Fig. 4.3, which is a 20.4 meter long Emmy Dyvi class ship and weighs 96 tons. The wave height at the time of the measurements was about 1 meter. The ship is owned by Redningsselskapet AS, Norway.

The most important information when including the future motion of the ship in a model predictive control approach is to have as accurate predictions as possible of the velocity and the acceleration entries of the state. If gravity plays an important role, the attitude of the ship should also be included. Fig. 4.4 shows an example of the true and estimated angular acceleration (roll) of the ship. Note that the predicted acceleration needs to be estimated at short time intervals to maintain a low prediction error. In general one can obtain very good results when the predictions are computed every 0.5 or 1.0 seconds. For predictions up to 3 seconds the predictions are also reasonable and no large errors occur. The prediction accuracy depends on the frequency of the waves. In general, predictions with horizons longer than one wave length, which in this case was 3-5 seconds, are not very reliable. This is mainly due to the fact that these may be out of phase which may lead to very large errors.

To get a more structured formulation of how the error changes with the prediction

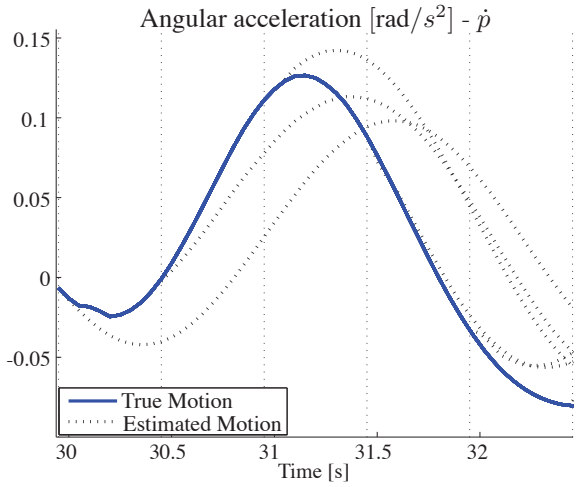


Figure 4.4: A typical example of true and predicted motion (AR) where a new prediction is calculated every 0.5 seconds. This clearly shows the need for re-computation of the predicted motion at short time intervals. The angular acceleration around the x-axis (roll) is shown.

horizon one can look at how the standard deviation evolves over time. This is important as it allows the path planner and controller to include uncertainty in the cost function and minimize this. Fig. 4.5 shows the standard deviation for the six degrees of freedom of the velocity state denoted $\nu = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^\mathsf{T}$. Note that the AR predictions are more accurate than the superposition of sines, except for horizons of 0.2 seconds or shorter. The AR method is thus chosen to predict the ship motions to be used in the motion planning and control presented in the next sections.

4.6.2 Experimental Setup

The experiments are performed using true motion data from the full-scale ship "Erik Bye" and predictions from the AR model as presented in Section 4.4. All the data is obtained using inertial sensing in six degrees of freedom. For the experimental setup a 9-DoF robot (3-DoF gantry crane and 6-DoF industrial manipulator, see Fig. 4.6) is used. A Güdel gantry crane with three translational degrees of freedom is used to hold an ABB IRB-2400 with six rotational degrees of freedom which is mounted upside down as in Fig. 4.6. The end effector of the robot is a camera which weighs about 1 kg. Due to limitations in the workspace of the gantry crane some of the components of the base motion had to be scaled down. The sway motion is scaled down with about 10% and the heave motion is scaled down with a factor of 5 to avoid collision between the manipulator and the floor.

The first 5-DoF were used to generate the ship motion. The robot thus generates the surge, sway and heave motion with the linear actuators of the gantry crane and the roll and pitch motion with the first two rotational joints of the manipulator. The yaw motion is very small and can be neglected. The last four links of the manipulator are then considered a standard manipulator on which the control and motion planning algorithms are tested.

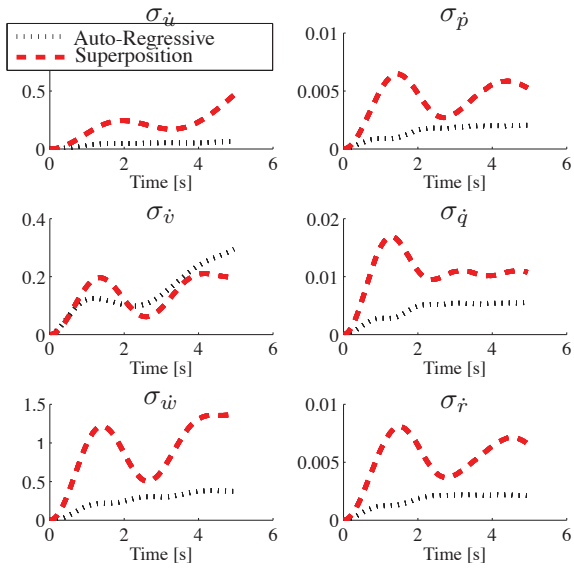


Figure 4.5: The standard deviation as a function of the prediction horizon. Note that the standard deviation is smaller with the AR model than for the superposition of sines, except for predictions of 0.2 seconds or less (not visible on the figure). The standard deviation is calculated from 200 samples, for "Erik Bye" moving at 15 knots at 1 meter wave height.

The motion of the "base", i.e., link 2 of the manipulator (see Fig. 4.6), is set to exactly the same as the measurements taken from the full-scale ship. This setup thus allows for very realistic experiments as the base link has exactly the same motion as the real ship. Due to the difference in the inertia between the ship and the manipulator the motion of the manipulator does not affect the motion of the ship. The dynamic coupling between the base link and the last four links of the manipulator is thus the same as for a 4-DoF manipulator mounted on a moving base.

4.6.3 Experimental Results Based on Predicted Ship Motions

This section shows how to exploit the inertial forces and choose a trajectory that minimizes the torques and the strain and tension on the manipulator. The approach is based on From et al. (2009a) and the motion planning problem is solved by numerically minimizing an objective function. Two different objective functions are used: i) the objective function in (4.17) which minimizes torques only, and ii) the objective function in (4.25) which minimizes torques and the variance.

For a given interval $(T_0, T_1) = (0, 10)$, the optimization problem is then to find the start and the end time for each joint with the restriction that $T_0 \leq t_{i,0} < t_{i,1} \leq T_1$ for all i. The shape, but not the starting time or length, of the trajectory is thus assumed fixed. By fixing the shape, a sub-optimal solution that can be solved efficiently and in

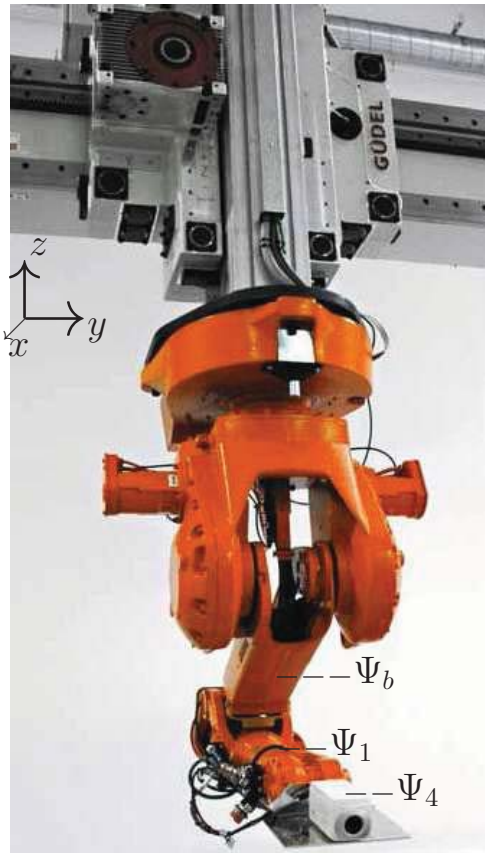


Figure 4.6: The ABB IRB-2400 robot mounted on a Güdel gantry crane. The first 5 joints (the 3 of the gantry crane and the first 2 of the robot) are used to generate the ship motion, represented by Ψ_b, and the last four joints of the robot, represented by $\Psi_{1,2,3,4}$, are used for optimal motion planning of a 4-DoF robot. Links 2 and 3 of the robot are hidden in the wrist and cannot be seen in the figure. Courtesy ABB Strategic R&D for Oil, Gas and Petrochemicals.

real time is found. Finding the optimal solution over all trajectories is a huge optimization problem and computationally too demanding to be solved in real time. The start and target configurations are chosen as

$$q(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}},$$
$$q(T) = \begin{bmatrix} \frac{\pi}{2} & \frac{\pi}{2} & \frac{\pi}{2} & \frac{\pi}{2} \end{bmatrix}^{\mathsf{T}}.$$

The motion planning problem thus amounts to finding the eight parameters (one start and end time for each joint) that minimize the cost integrated over a fixed time interval while the robot start and target configurations are satisfied.

For the first cost function (4.17) let D be a positive definite matrix that defines a metric in τ-space. Choose $D = 10^{-6} \cdot \operatorname{diag}\begin{bmatrix} 10 & 2 & 2 & 5 \end{bmatrix}$ reflecting the masses of the links.

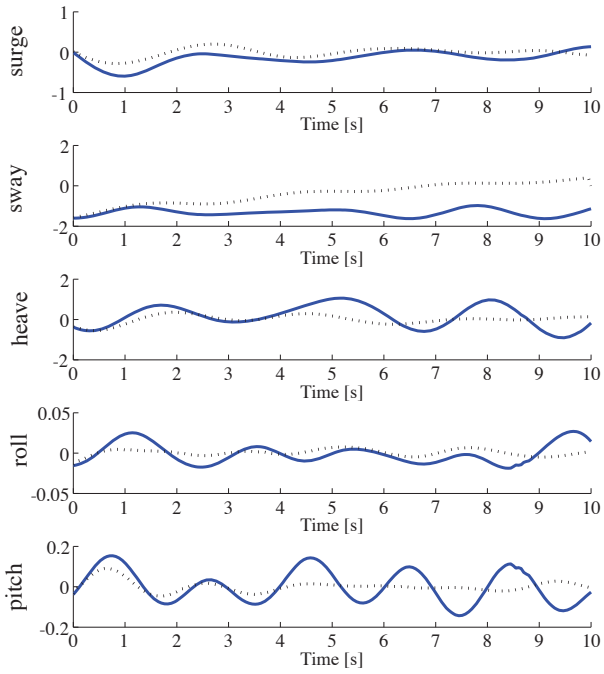


Figure 4.7: The motion collected from the ship measurements and given as input to the first five joints of the manipulator. The full line shows the measured accelerations and the dotted line shows the predicted ship motion used in the motion planners. Note that the predictions have a damping effect as the predictions become less accurate.

For the second cost function (4.25) the weight is chosen as $\kappa = 10$ to enhance the effect of adding the variance, but otherwise the same weighing as for (4.17) is used.

First the experiments are performed with a baseline trajectory, i.e., a starting time $t_{i,0} = T_0$ and end time $t_{i,1} = T_1$ for all i. This is how one would choose the trajectory when no information about the motion of the ship is available. Second, the optimization problem is solved assuming complete knowledge about the ship motion, including future motion. This gives information about how well one can expect to perform if accurate motion predictions are available. Finally, two experiments where the trajectory is calculated based on the predicted base motion but the base motion is exactly the motion of the ship are performed. This is done with the cost function given in (4.17) and in (4.25). Thus, the four algorithms in Table 4.1 were tested.

At the beginning of the time interval the next 10 seconds of ship motion are predicted based on the previous 5 seconds of known ship motion. Then, after 5 seconds the current position of the ship is updated and the ship motion for the next 5 seconds is predicted. Thus, the prediction horizon is 10 seconds for the first half of the time interval, and 5

Alg. 1 - (— —)	A baseline trajectory with maximum motion duration for all the joints.
Alg. 2 - (· · · · ·)	An optimized trajectory taking the correct base motion into account.
Alg. 3 - (- - - -)	An optimized trajectory taking the predicted base motion into account with cost function (4.17).
Alg. 4 - (— · —·)	An optimized trajectory taking the predicted base motion into account with cost function (4.25).

Table 4.1: The four algorithms studied. Note the colour coding used in the figures in this section.

seconds for the last half. After 5 seconds the optimal trajectory is thus re-computed based on the new information and the new trajectory is followed. As accurate predictions are available for about 3 seconds the algorithms must thus handle the fact that for part of the interval, the predictions are not very reliable.

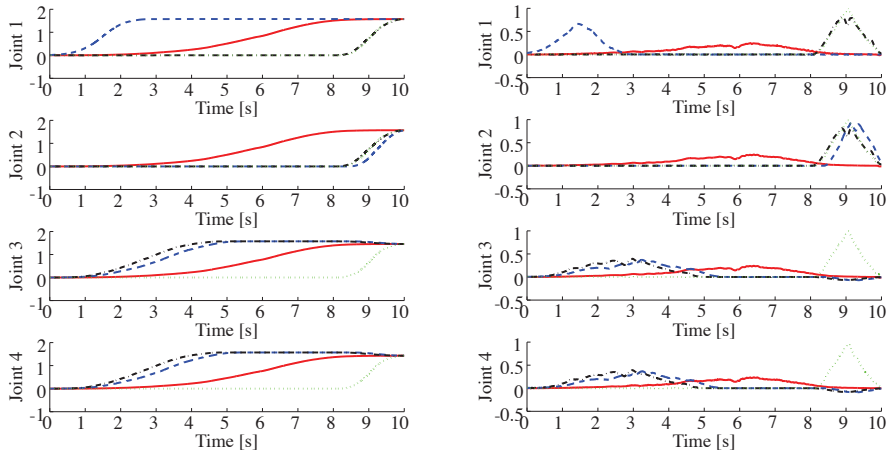
To compare the energy used for the different algorithms the following cost function (power) is used:

$$P_i = \int_{t=T_0}^{T_1} \tau_i \dot{q}_i dt \tag{4.27}$$

for each joint $i = 1 \dots 4$. This gives a good indication of how much energy is used to take the manipulator from the start to the target configuration.

The positions and the velocities of the four joints for the four different trajectories are found in Fig. 4.8. Note that the trajectories found for the four different approaches are quite different. First, the trajectories based on the true and predicted base motions are quite different, especially for joints 3 and 4. The trajectory based on the true ship motion (Alg. 2) takes advantage of a favourable motion at the end of the interval while the trajectories based on the predicted motion (Alg. 3 and 4) take advantage of a favourable motion in the beginning of the interval. This difference arises because the ship motion towards the end of the interval is not available to the algorithms based on the predicted ship motion, due to the long horizon and the damping effect seen in Fig. 4.7. Alg. 2 chooses to wait to start the motion until the very last minute. This probably has a simple explanation. All the joints start out in the initial condition, as shown in Fig. 4.6, and end up with a rotation of $\frac{\pi}{2}$ radians for all the joints. The largest components of the ship motion lie in the xz-plane, i.e., the pitch is far bigger than the roll, and the sway is fairly constant at -1 meter. It can be seen from Fig. 4.6 that the inertial forces affect the manipulator less for the initial configuration than for the target configuration and thus less energy is used if the motion is delayed until the end of the interval.

There is also an interesting difference between the torque-based cost function (Alg. 3) and the cost function that is based on both torque and variance (Alg. 4). Note that Alg 4. calculates an optimal trajectory that is closer to Alg. 2, which is considered the optimal trajectory. This can be seen in joint 1, which is the most important joint due to its inertia and also the joint with the highest weight in the cost function. As the uncertainty



(a) Joint Positions.

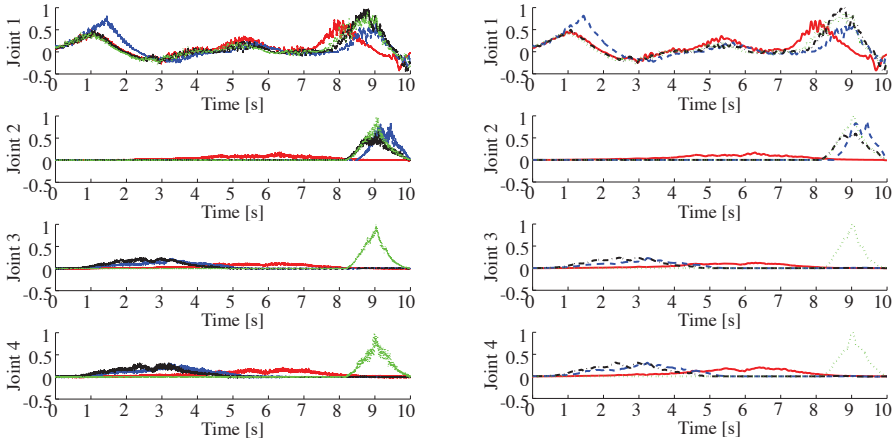
(b) Joint Velocities.

Figure 4.8: Four different trajectories are shown: Alg. 1 - a baseline trajectory with maximum motion duration (full, red lines), Alg. 2 - an optimal trajectory taking the correct base motion into account (dotted, green lines), Alg. 3 - an optimized trajectory taking the predicted base motion into account (dashed, blue lines), and Alg. 4 - an optimized trajectory taking the predicted base motion into account where the variance is included in the cost function (dot-dashed, black lines).

is biggest in surge, heave, and pitch, and the variance is included in the cost function, this algorithm will try to keep a configuration for which these components of the ship motion do not affect the manipulator. In other words, the algorithm will choose a trajectory where the mapping from the axes with the largest uncertainties (surge, heave, and pitch) to the internal forces of the manipulator is as small as possible, which is the reason for delaying the motion for joint 1.

Fig. 4.9 shows the value of the cost (4.27) for the four algorithms, i.e., torque times velocity for each joint. The most apparent observation here is the difference between joint 1, which is a very heavy joint, and joints 2, 3 and 4, which are very light, i.e., about 10% of the weight of joint 1. Note that due to the motion of the base, joint 1 needs to use a substantial amount of torque just to keep the arm fixed. In fact, the largest amount of torque is used to compensate for the base motion, and not to move the joint from the start to the target configuration. This is a surprising result as the base motion simulates only 1 meter wave height and that some components of the motion are scaled down. One can thus conclude that for a manipulator mounted on a ship in high sea, the amount of torque needed to compensate for the inertial forces will be substantial and needs to be included in both the motion planning and control of the manipulator in order to obtain accurate and energy efficient control.

Table 4.2 shows the values of the cost function (4.27) for each joint for the four algorithms studied. The average values of the cost function (4.27) for all joints can be found in Table 4.3. All the entries are scaled so that the maximum value equals 1, so the values of



(a) Measured joint torques times velocities.

(b) Low pass filtered torques times velocities.

Figure 4.9: The torques times velocity over the trajectory. Four different trajectories are shown: Alg. 1 - a baseline trajectory with maximum motion duration (full, red lines), Alg. 2 - an optimal trajectory taking the correct base motion into account (dotted, green lines), Alg. 3 - an optimized trajectory taking the predicted base motion into account (dashed, blue lines), and Alg. 4 - an optimized trajectory taking the predicted base motion into account where the variance is included in the cost function (dot-dashed, black lines).

the different joints cannot be compared directly. Note that all the optimization algorithms perform better than the baseline trajectory, in terms of power. Also note that the optimization based on the predicted base motion performs better than the optimization based on the real base motion when evaluating the cost given in (4.27), which may come as a surprise.

Joint	Alg. 1	Alg. 2	Alg. 3	Alg. 4
1	1.000	0.787	0.627	0.856
2	1.000	0.061	0.036	0.817
3	1.000	0.652	0.331	0.960
4	1.000	0.709	0.657	0.669

Table 4.2: Square of the torque times velocity over the trajectory for each joint.

	Alg. 1	Alg. 2	Alg. 3	Alg. 4
Average	1.0000	0.552	0.413	0.825

Table 4.3: Average of the square of the torque times velocity over the trajectory.

On the other hand, as can be seen in Table 4.4, this is not the case when the square of the torque is used. In this case the optimization based on the real base motion performs best, as should be expected as this is the cost function used in the optimization algorithm. Also note that the cost with torque only performs better than when the variance is added. This is also as should be expected as the cost function used in Alg. 3 is the one found in (4.27), just with different weights for the four joints.

Joint	Alg. 1	Alg. 2	Alg. 3	Alg. 4
1	1.000	0.807	0.844	0.948
2	1.000	0.292	0.417	0.292
3	1.000	0.756	0.511	0.689
4	1.000	0.571	0.327	0.612

Table 4.4: Square of the torques over the trajectory for each joint.

	Alg. 1	Alg. 2	Alg. 3	Alg. 4
Average	1.000	0.789	0.799	0.910

Table 4.5: Average values of the square of the torques over the trajectory.

Tables 4.6 and 4.7 show the maximum torques for the four algorithms. In this case, Alg. 4 performs better than Alg. 3. This was the main intention of adding the variance to the cost function. When the variance is added to the cost function, the algorithm (Alg. 4) chooses a safer path in the sense that the mapping from the uncertainties of the base motion to the uncertainty in the robot state will minimize the uncertainty in the robot state. Also note, however, that the safest path in terms of maximum torques seems to be the one with the longest duration and thus more evenly distributing the torques over the entire interval.

Joint	Alg. 1	Alg. 2	Alg. 3	Alg. 4
1	0.945	1.000	0.968	0.935
2	0.691	0.876	0.931	1.000
3	0.706	0.768	1.000	0.666
4	0.783	0.762	1.000	0.824

Table 4.6: Maximum of the torques for each joint.

	Alg. 1	Alg. 2	Alg. 3	Alg. 4
Average	0.873	0.935	1.000	0.903

Table 4.7: The average of the maximum of the torques.

4.6.4 Simulations Based on Predicted Ship Motions

Simulations are used to verify the experimental studies by running the motion planning algorithm several times based on different motion prediction data sets. 200 data sets that all have the same characteristics were chosen in order to be able to draw some general conclusions and compare the results to the ones found in experiments. All the sets are picked from one long sampling and are thus collected during a short period of time and in a sea state for which the dominant components of the motion were in the xz-plane, i.e., the pitch is far bigger than the roll, and the sway is almost zero. In other words, all the data sets are measurements of the ship moving with the same velocity, in the same sea state, and with the same attack angle on the waves.

Table 4.8 confirms the tendency from Table 4.5: all the algorithms perform better than the benchmark solution. In fact, all approaches obtain better results in the simulations than for the experiments. This is a promising result as the large number of simulations make these results meaningful also for stochastic data. Note that the approach based on the real base motion performs better in this case. This is reasonable as the amount of accurate data available should allow for a more optimal solution to be found. Thus, the small difference between Alg. 2 and 3 in Table 4.5 cannot be taken as a general result.

	Alg. 1	Alg. 2	Alg. 3	Alg. 4
Average	1.000	0.670	0.788	0.829

Table 4.8: The square of the torques over the trajectory for all the joints. Average values of all the joints for 200 samples.

Also, the maximum values found in Table 4.7 can be confirmed. However, Table 4.9 shows that Alg. 2 performs better than Alg. 1, which is not the case for the experiments. Once again the cost function based on both the torque and the variance decreases the maximum values of the torque, but increases the overall torque used during the trajectory, which again is as expected.

	Alg. 1	Alg. 2	Alg. 3	Alg. 4
Average	0.890	0.823	1.000	0.888

Table 4.9: The maximum of the torques over the trajectory for all the joints. Average values of all the joints for 200 samples.

4.6.5 Horizon Length

This section discusses how to choose the horizon and the frequency at which the problem is re-solved in a receding horizon setting when only predictions of the base motions are known. The previous sections found that the predictions are very accurate for about 0.5 seconds and relatively accurate for a horizon of about 3 seconds. When it comes to motion planning, however, it turns out that a longer horizon should be chosen. The reason for this

is that the algorithm searches for the time interval in which the inertial forces contribute to the motion as much as possible. It is thus desirable to choose the horizon as long as possible, but at the same time avoid using predictions that are not accurate and may lead to large errors. Assume the horizon is equal to the re-computation interval. Table 4.10 shows the optimal horizon/re-computation interval, the total cost and the maximum torques for i) a cost function minimizing the torques only (Alg. 3), and ii) a cost function minimizing the torques and variance (Alg. 4). Table 4.10 shows the value of the cost function based on only the torque for both algorithms in order to be able to compare the two values.

When the variance is included in the cost function one can allow a longer horizon. This allows the path planner to use more of the information available, albeit inaccurate, and results in a slightly lower cost than when only the torque is used. The advantage of adding the variance is rather small when looking at the squared torque, so the main advantage of including the variance in the cost function is that maximum values of the torque decreases, as seen in Table 4.9.

Horizon	Torque	
[s]	Alg. 3	Alg. 4
1	1.000	0.897
3	0.812	0.688
4	0.630	0.688
5	0.667	0.670
6	0.778	0.611
7	1.287	0.687

Table 4.10: The cost for different horizons for a cost function without and with variance, based on 200 simulations.

4.7 Related Research

Research on several related topics can be found in the literature. Love et al. (2004) address the impact of wave generated disturbances on the tracking control of a manipulator mounted on a ship based on the classical Lagrangian approach. They use repetitive learning control and this results in performance improvement for purely periodic motions, but they do not present a formal derivation of the dynamics equations. Kitarovic et al. (2005) and Oh et al. (2005) address the use of cable robots for loading and unloading cargo between two ships. In the Ampelmann project (Salzmann, 2007), a Stewart platform is mounted on a ship and is used to compensate for the motion of the ship by keeping the platform still with respect to the world frame. Lebans et al. (1997) give a cursory description of a telerobotic shipboard handling system, and Kosuge et al. (1992) and Kajita and Kosuge (1997) address the control of robots floating on the water utilizing vehicle restoring forces. Other related research areas are macro/micro manipulators (Yoshikawa et al., 1996) and (Bowling and Khatib, 1997), underwater vehicle/manipulator systems (McMillan et al., 1995) and spacecraft-manipulator systems (Egeland and Sagli, 1993).

Most previous work deals with robots mounted on a free-floating base. There is, however, an important difference between modeling a robot on a forced and a free-floating

base. A forced base motion will add inertial forces to the dynamic equations that do not arise in the free-floating case, such as spacecraft-manipulator systems and manipulators on small AUVs.

There are some papers in the literature considering the prediction of ship motion. Yang et al. (2008a,b) discuss the problem of landing a helicopter on a ship in high sea and predict the ship motion by fitting the ship model to the measured data using recursive least squares. Khan et al. (2005) use artificial neural networks to solve the same problem. In Chung et al. (1990) the sea excitation is extrapolated using the superposition of sines approach and the ship motion is predicted using the ship model driven by the extrapolated forces.

Stochastic model predictive control (MPC) is discussed in detail in Cannon et al. (2007) and Couchman et al. (2006) where the output variance is included in the cost function. The relation between the input and the output variance is also important in performance assessment, which is discussed in an MPC setting in Zhang and Li (2007) and the minimum variance performance map is discussed in Harrison and Qin (2009). The output variance is also discussed in the minimum variance control of stochastic processes developed by Astrom in the 1960s (Åstrøm, 1967).

4.8 Conclusions

This paper presents the first detailed discussion regarding several important aspects of ship-manipulator systems. First, the extent to which the ship motion can be predicted and for what horizon this can be used in the motion planning and control of the manipulator is investigated. Then several different approaches to the motion planning problem for robots mounted on ships are discussed and it is shown that the amount of torque needed to reach a desired configuration can be reduced by including the predicted base motion in the motion planner. The torques needed to reach the target can be reduced even for relatively moderate ship motions. Thus, one may conclude that for a ship in high sea it is possible to substantially improve performance and allow for efficient motion planning and accurate control in settings where this would otherwise not be possible due to large inertial forces. It is also shown that by including the variance of the predicted motion for the different degrees of freedom in the motion planner the maximum torques needed and thus also the strain and tension on the robot are reduced.

Several simulations and experiments are performed to validate the approaches presented. For the first time, detailed experimental results of ship-manipulator systems under the influence of inertial forces are presented. Several simulations with a large number of data sets are also performed and it is shown that the simulation results are consistent with the experimental result. Also for the first time, the effect of including the variance in the cost function in a receding horizon control law is investigated through experiments based on real ship motion data.

It is found that the inertial forces that act on a manipulator mounted on a ship in only 1 meter wave height pose significant challenges in robot control and motion planning, especially for the joints with large inertia. Also, for a ship in high sea state, the inertial forces are significantly higher and must be included in both the motion planner and the control to guarantee safe, cost efficient and accurate manipulation. An interesting topic for future work is thus to repeat the same experiments for a ship in high sea.

Chapter 5

On the Boundedness Property of the Inertia Matrix and Skew-Symmetric Property of the Coriolis Matrix for Vehicle-Manipulator Systems

5.1 Abstract

This paper addresses the boundedness property of the inertia matrix and the skew-symmetric property of the Coriolis matrix for vehicle-manipulator systems. These properties are widely used in control theory and Lyapunov-based stability proofs, and are therefore important to identify. For example, the skew-symmetric property does not depend on the system at hand, but on the choice of parameterisation of the Coriolis matrix, which is not unique. It is the authors' experience that many researchers take this assumption for granted without taking into account that there exist several parameterisations for which this is not true. In fact, most researchers refer to references that do not show this property for vehicle-manipulator systems, but for other systems such as single rigid bodies or manipulators on a fixed base. As a result, the otherwise rigorous stability proofs fall apart. In this paper we point out several references that are widely used, but that do not show this property and we refer to the correct references. As most references on this topics are not easily accessible, we also give the correct proofs for commonly used parameterisations of the Coriolis matrix and thus provide a proof for future reference. We also correct several mistakes made in the proofs of the aforementioned references.

The same is the case for the boundedness property of the inertia matrix which for a bad choice of state variables will not necessarily hold. This is an important property in several different control laws, such as robust control, and also in simula-

tion of vehicle-manipulator systems. These control laws can be shown to be globally valid for single rigid bodies or fixed base manipulators, but not for the most common formulations of vehicle-manipulator systems which are singularity prone. This can be solved by deriving the dynamics in terms of quasi-velocities, which allows us to describe the dynamics without the presence of the Euler angle singularities that normally arise in vehicle-manipulator dynamics. When the singularities are removed from the equations, we get an inertia matrix that is bounded in its variables.

To the authors' best knowledge we derive for the first time the dynamic equations with both the skew-symmetric property of the Coriolis matrix and the boundedness property for the inertia matrix for vehicle-manipulator systems with non-Euclidean joints.

Keywords: Vehicle-manipulator dynamics, robot modelling, dynamic properties, singularities.

5.2 Introduction

This paper is motivated by a general concern that some frequently used properties of the inertia and Coriolis matrices for vehicle-manipulator systems are assumed true based on the proofs for other systems. We show that the proofs of these properties for fixed-base robot manipulators or single rigid bodies (vehicles) cannot be generalised to vehicle-manipulator systems directly. In fact, we show that the most commonly used dynamic equations for vehicle-manipulator systems do not possess both the boundedness and skew-symmetric properties. There is thus a need to clarify to what extent these properties are true and to find a rigorous mathematical representation of these systems for use in simulations and controller design. To this end we present a reformulation of the dynamic equations for vehicle-manipulator systems for which both the boundedness and the skew-symmetric properties are true.

Lyapunov based controllers are based on several assumptions that make the controller design both more convenient and physically meaningful. These assumptions should thus reflect the physical properties of the system. With the increasing popularity of Lyapunov design some of these properties are almost universally taken for granted. In this paper we discuss two such properties that cannot be assumed true for vehicle-manipulator systems without further consideration.

The first property is concerned with the boundedness of the inertia matrix M, i.e. if there exist lower and upper bounds on its singular values. For robot manipulators this boundedness property is addressed in Ghorbel et al. (1998) where the class of robots for which the inertia matrix is bounded is characterised. The work of Ghorbel et al. (1998) differs from our approach in that they are mainly concerned with the design of the manipulator while we are concerned with the mathematical representation. For a given robotic manipulator there may exist one mathematical representation for which the inertia matrix is bounded and another for which it is not. We find that for the most commonly used mathematical representation of vehicle-manipulator systems this property is not true.

The second property that we are concerned with is to find a parameterisation of the Coriolis matrix C so that the matrix $\dot{M} - 2C$ is skew-symmetric. Such a parameterisation is easy to find for fixed-base robots or for single rigid bodies, but not always for vehicle-

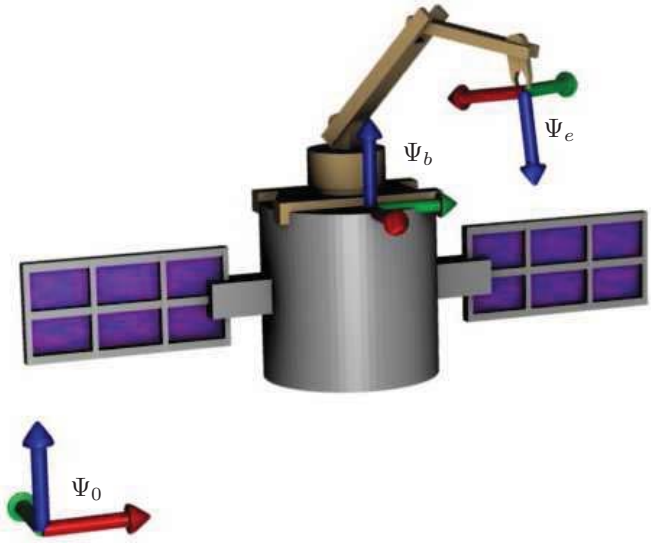


Figure 5.1: Model setup for a four-link robot attached to a non-inertial base with coordinate frame Ψ_b. Frame Ψ_0 denotes the inertial reference frame and Ψ_e is the end-effector frame.

manipulator systems. Particularly we find that such a parameterisation is hard to find, especially together with the boundedness property.

We will focus on two important classes of vehicle-manipulator systems—underwater-manipulator systems and spacecraft-manipulator systems—but the results are general and also applicable to other vehicles. Underwater-manipulator systems are extensively treated in Antonelli (2006), Schjølberg (1996), Schjølberg and Fossen (1994) and Fossen and Fjellstad (1995). For the choice of state variables used in most of the literature, the boundedness property does not hold for the whole configuration space, i.e. there exist isolated points where the inertia matrix becomes singular. This can, however, be dealt with by introducing a quaternion representation (Fossen, 2002). The quaternion representation is well suited for single rigid bodies, but for multibody systems the Euler angles are normally adopted. The problems regarding the Euler angle singularities are pointed out in most books and papers when it comes to modelling, but is often left out when dealing with stability proofs. As a result of this the control law is not valid at isolated points in the configuration space.

Similarly, the skew-symmetric property of the Coriolis matrix is in general not treated correctly and is in most cases assumed true without any further proof. In the authors' view, this is a strong weakness because this property depends on how we choose to represent the Coriolis matrix. It is thus not sufficient to refer to an arbitrary proof of skew-symmetry: one must refer to a proof for the specific parameterisation of the Coriolis matrix chosen. Most papers on the topic refer to Antonelli (2006), Fossen and Fjellstad (1995), de Wit et al. (1998) or Schjølberg and Fossen (1994) for this proof. However, none of these references actually show the proof. Given the velocity state v, Schjølberg and Fossen (1994) state that $v^\mathsf{T}(\dot{M} - 2C)v = 0$, which is true, but a weaker result than skew-symmetry.

This property is known as the *principle of conservation of energy* and is always true. This is often used to show skew-symmetry, which is not correct. Other commonly used references are taken from the fixed-base robotics literature, such as Murray et al. (1994) and Sciavicco and Siciliano (2005). The proof can be found in Schjølberg (1996), but only for systems where the boundedness property does not hold. We present this proof, and correct some mistakes made, so that this proof is correctly presented for future reference.

Spacecraft are normally modelled using quaternions and the inertia matrix is thus bounded for the whole configuration manifold (Wen and Kreutz-Delgado, 1991). A Lie group formulation of the dynamics of a rigid body is studied in Bullo and Lewis (2004) and Marsden and Ratiu (1999). For spacecraft-manipulator systems, however, a Langrangian approach is normally adopted and, again, the dynamics are not globally valid. Such systems are discussed in Hughes (2002), Moosavian and Papadopoulos (2007), Liang et al. (1997, 1998) and Vafa and Dubowsky (1987). As for the underwater systems, most papers concerned with modelling address the boundedness property, but it is often not noted in the stability proofs (Antonelli, 2006). Also for the skew-symmetric property the most commonly used references only show this property for fixed-base manipulators, such as in Murray et al. (1994), Sciavicco and Siciliano (2005) and Craig (1987). For the formulation presented in Egeland and Pettersen (1998) the dynamics possess the skew-symmetric property and, based on the proof in Schjølberg and Fossen (1994), we show that this can be shown also when the dynamics are written in terms of global state variables.

5.3 Properties of the dynamics

In this section we list some important properties of dynamical systems in matrix form that play important roles in system analysis as well as controller design. Assume for now that we can write the dynamic equations of a mechanical system in the form

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau \tag{5.1}$$

where q is the state of the system, $M(q)$ is the inertia matrix and $C(q,\dot{q})$ is the Coriolis and centripetal matrix. The following properties can be associated with the inertia and Coriolis matrices (Børhaug, 2008):

Property 1. (The boundedness property) The inertia matrix $M(q)$ is uniformly bounded in q, i.e. there exist constants d_1 and d_2, such that

$$0 < d_1 \leq ||M(q)|| \leq d_2 < \infty, \qquad \forall\, q \in \mathbb{R}^n, \tag{5.2}$$

where $||\cdot||$ is the induced norm for matrices, i.e. a max-bound on the maximum singular value and a min-bound on the minimum singular value of the matrix.

Property 2. (The skew-symmetric property) The matrix $(\dot{M}(q) - 2C(q,\dot{q}))$ is skew-symmetric.

Property 1 is true only when there are no singularities present in the formulation. Thus, if the Euler angles are used to represent the attitude of the vehicle, as in Fossen (2002), Schjølberg (1996) and Børhaug (2008), this is not satisfied. The existence of the boundaries d_1 and d_2 is the basis of gain controller design and global Lyapunov stability, and

is used in several manipulator control laws such as robust control (Ghorbel et al., 1998; Sciavicco and Siciliano, 2005). Given a computed estimate of the inertia matrix denoted $\hat{M}$ many controllers assume the property

$$\left|\left|M(q)^{-1}\hat{M}(q) - I\right|\right| \leq d < 1, \qquad \forall\, q \in \mathbb{R}^n \tag{5.3}$$

which is automatically satisfied if the constant d is chosen as

$$d = \frac{d_2 - d_1}{d_2 + d_1}. \tag{5.4}$$

Property 1 guarantees that the constant d is bounded and is thus important in a large class of existing control laws. We will see one such example in Section 5.4.1.

Property 2 is true for a certain parameterisation of the Coriolis matrix. Such a representation is well known for robotic manipulators on a fixed base (Murray et al., 1994; Sciavicco and Siciliano, 2005) and for vehicles with no manipulator attached (Fossen, 2002). One formulation for spacecraft-manipulator systems is found in Egeland and Pettersen (1998) where the boundedness and skew-symmetric properties are both true. The formulation uses quasi-velocities and the final equations resemble Kirchhoff's equations (Fossen, 2002), but for multibody systems.

The formulation presented in Egeland and Pettersen (1998) is, however, independent of the vehicle configuration of the vehicle, and there is no obvious way to include these without introducing the singularities that arise in the transformation between the local and global velocity variables. The configuration of the vehicle is important in order to include terms such as gravity and buoyancy in the dynamics. Also other forces, such as ocean currents are added to the dynamics considering the relative velocity ν_{rel} in body-fixed frame, i.e.

$$\nu_{rel} = \nu - R_{0b}\nu_{curr} \tag{5.5}$$

where ν is the linear and angular velocity in the body frame, ν_{curr} is the velocity of the ocean current given in the inertial frame and R_{0b} is the rotation matrix. Hence, this transformation also requires the position state of the vehicle.

The formulation found in Egeland and Pettersen (1998) does not need the position variables of the vehicle because they consider this the first link. For more general systems the position variables may be required in the inertia matrix, and in this setting the formulation in Egeland and Pettersen (1998) does not hold in terms of the boundedness property. This includes systems with more than one transformation that cannot be represented with generalised coordinates.

Based on these observations, we organise the paper as follows: In the next section we start with a motivational example and show how these properties appear in control algorithms of mechanical systems. In Section 5.5 we derive the dynamics of Lagrangian systems in terms of generalised coordinates which include fixed-base robotic manipulators with 1-DoF revolute or prismatic joints and we show that both Properties 1 and 2 hold. In Section 5.6 we derive the dynamics for single rigid bodies/vehicles when the state space cannot be written in terms of generalised coordinates and we show that Property 2 holds, but Property 1 does not. Section 5.7.1 describes the dynamics of vehicle-manipulator systems as they normally appear in the literature and we show that the boundedness property

does not hold. In Section 5.7.2 the dynamics of vehicle-manipulator systems are derived based on Egeland and Pettersen (1998) and the boundedness and skew-symmetric properties are shown. We also correct some mistakes that occur in Egeland and Pettersen (1998). In Section 5.7.3 we present the correct equations based on their approach and extend these to the more general case when also the position variables of the vehicle are included. We also present the explicit expressions of the matrices that do not appear in Egeland and Pettersen (1998) as well as a detailed proof of the skew-symmetric property. In Section 5.7.3 we present for the first time a set of equations where both the boundedness and skew-symmetric properties hold for multibody systems that cannot be written in terms of generalised coordinates.

5.4 The Boundedness and Skew Symmetric Properties in Control

In this section we illustrate how the boundedness property of the inertia matrix appears in control schemes such as robust control and how the skew-symmetric property of the Coriolis matrix appears in the stability proof of PD control laws. The next two sections are based on the control laws presented in Sciavicco and Siciliano (2005) and Murray et al. (1994), respectively.

5.4.1 Robust Control

Assume the dynamics of a robotic manipulator written in the form of (5.1) with the control

$$\tau = \hat{M}(q)y + \hat{C}(q,\dot{q})\dot{q} \tag{5.6}$$

where $\hat{M}$ and $\hat{C}$ represent the computed estimates of the dynamic model. We choose the control action as

$$y = \ddot{q}_d + K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q) \tag{5.7}$$

where q_d is the desired trajectory in joint space. We further choose K_D and K_P as positive definite matrices which guarantees that the error converges to zero. We see this if we combine (5.1) and (5.6) which gives

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = \hat{M}(q)y + \hat{C}(q,\dot{q})\dot{q}. \tag{5.8}$$

The inertia matrix of standard industrial manipulators is invertible and we can rewrite this as

$$\ddot{q} = y + (M^{-1}(q)\hat{M}(q) - I)y + M^{-1}(q)(\hat{C}(q,\dot{q}) - C(q,\dot{q}))\dot{q}. \tag{5.9}$$

Thus, as we would like to have $\ddot{q} = y$, the uncertainty is represented by

$$\gamma = (I - M^{-1}(q)\hat{M}(q))y - M^{-1}(q)(\hat{C}(q,\dot{q}) - C(q,\dot{q}))\dot{q}. \tag{5.10}$$

We can now write the error dynamics as

$$\ddot{e} + K_D\dot{e} + K_P e = \gamma \tag{5.11}$$

where $e = q_d - q$ and the left hand side guarantees convergence to zero.

The right hand side of (5.11) represents the uncertainties of the system. We need to find a control law that guarantees asymptotic stability given that an estimate of the range of variation of the uncertainties is available. Based on (5.10) we can set up the following assumptions that will guarantee asymptotic stability (Sciavicco and Siciliano, 2005)

$$\sup_{t\leq 0} ||\ddot{q}_d|| < d_M < \infty \qquad \forall \ddot{q}_d, \tag{5.12}$$

$$\left|\left| I - M^{-1}(q)\hat{M}(q) \right|\right| \leq d < 1 \qquad \forall q, \tag{5.13}$$

$$\left|\left| \hat{C}(q,\dot{q}) - C(q,\dot{q}) \right|\right| \leq d_\phi < \infty \qquad \forall q, \dot{q}. \tag{5.14}$$

Assumption (5.12) is trivially satisfied as our planned trajectory should not require infinite accelerations. Assumption (5.14) depends only on q and $\dot{q}$, and is satisfied if we assume that the joint ranges are limited and that there exist maximum saturations on the velocities of the motors, which is the case in most mechanical systems and certainly for standard robotic manipulators.

Of special interest in the setting of this paper is Assumption (5.13). For this to be true, we need to guarantee that the inequality

$$0 < d_1 \leq \left|\left| M^{-1}(q) \right|\right| \leq d_2 < \infty, \qquad \forall\, q \in \mathbb{R}^n. \tag{5.15}$$

holds. If $M(q)$ is bounded with lower and upper bounds, this inequality holds and we can always find a matrix $\hat{M}(q)$ that satisfies (5.13). For example, if we set

$$\hat{M} = \frac{2}{d_2 + d_1} I \tag{5.16}$$

where I is the identity matrix we get

$$\left|\left| M(q)^{-1}\hat{M} - I \right|\right| \leq \frac{d_2 - d_1}{d_2 + d_1} < 1, \qquad \forall\, q \in \mathbb{R}^n \tag{5.17}$$

and (5.13) is satisfied. Hence, the property that the inertia matrix is strictly positive definite and bounded is used explicitly in the stability proofs.

5.4.2 PD Control Law

Stability in the sense of Lyapunov can be summarised in the following way: *Let $V(x,t)$ be a non-negative function with derivative $\dot{V}$ along the trajectories of the system. If $V(x,t)$ is locally positive definite and $\dot{V}(x,t) \leq 0$ locally in x and for all t, then the origin of the system is locally stable.*

We will now see an example of how the skew-symmetric property of the Coriolis matrix plays an important role in Lyapunov-based stability proofs. Assume the system (5.1) and the augmented PD control law

$$\tau = M(q)\ddot{q}_d + C(q,\dot{q})\dot{q}_d - K_D\dot{e} - K_P e \tag{5.18}$$

where $e = q_d - q$. The closed loop system is then given by

$$M(q)\ddot{e} + C(q,\dot{q})\dot{e} - K_D\dot{e} - K_P e = 0. \tag{5.19}$$

To show stability we choose the Lyapunov function

$$V(e,\dot{e},t) = \frac{1}{2}\dot{e}^\mathsf{T} M(q)\dot{e} + \frac{1}{2}e^\mathsf{T} K_P e + \epsilon e^\mathsf{T} M(q)\dot{e} \tag{5.20}$$

which is positive definite for sufficiently small ϵ. We now evaluate $\dot{V}$ along the trajectories of (5.19):

$$\begin{aligned}\dot{V} =& \dot{e}^\mathsf{T} M\ddot{e} + \frac{1}{2}\dot{e}^\mathsf{T}\dot{M}\dot{e} + \dot{e}^\mathsf{T} K_P e + \epsilon\dot{e}^\mathsf{T} M\dot{e} + \epsilon e^\mathsf{T}(M\ddot{e} + \dot{M}\dot{e}) \\ =& -\dot{e}^\mathsf{T}(K_D - \epsilon M)\dot{e} + \frac{1}{2}\dot{e}^\mathsf{T}(\dot{M} - 2C)\dot{e} + \epsilon e^\mathsf{T}(-K_P e - K_D\dot{e} - C\dot{e} + \dot{M}\dot{e})\end{aligned} \tag{5.21}$$

For standard robotic manipulators—but, as we will see, not for general mechanical systems—we have $x^\mathsf{T}(\dot{M} - 2C)x = 0$, $\forall x$ and the second term vanishes. Similarly, the third term can be written as

$$\begin{aligned}e^\mathsf{T}(-K_P e - K_D\dot{e} - C\dot{e} + \dot{M}\dot{e}) &= -e^\mathsf{T} K_P e - e^\mathsf{T} K_D\dot{e} + e^\mathsf{T}(\dot{M} - C\dot{e})\dot{e} \\ &= -e^\mathsf{T} K_P e - e^\mathsf{T} K_D\dot{e} + e^\mathsf{T}(\frac{1}{2}\dot{M} - C\dot{e})\dot{e} + \frac{1}{2}e^\mathsf{T}\dot{M}\dot{e} \\ &= -e^\mathsf{T} K_P e - e^\mathsf{T} K_D\dot{e} + \frac{1}{2}e^\mathsf{T}\dot{M}\dot{e}\end{aligned} \tag{5.22}$$

and again we have used the skew-symmetric property of $(\dot{M} - 2C)$. We can now write (5.21) as

$$\begin{aligned}\dot{V} =& -\dot{e}^\mathsf{T}(K_D - \epsilon M)\dot{e} + \frac{1}{2}\dot{e}^\mathsf{T}(\dot{M} - 2C)\dot{e} + \epsilon e^\mathsf{T}(-K_P e - K_D\dot{e} - C\dot{e} + \dot{M}\dot{e}) \\ =& -\dot{e}^\mathsf{T}(K_D - \epsilon M)\dot{e} - \epsilon e^\mathsf{T} K_P e + \epsilon e^\mathsf{T}(-K_D + \frac{1}{2}\dot{M})\dot{e}\end{aligned} \tag{5.23}$$

which, for sufficiently small $\epsilon < 0$ guarantees that $\dot{V}$ is negative definite and that the system is exponentially stable. The stability proof thus requires that the terms with $(\dot{M} - 2C)$ vanishes.

We note that the stability proofs of neither the robust controller nor the PD controller are valid if Properties 1 and 2 are not true. With this as motivation we now investigate the validity of these properties for different formulations of the dynamic equations in matrix form for mechanical systems.

5.5 Lagrangian Dynamics on $\mathbb{R}^n$

A wide range of dynamical systems can be described by the Lagrange equations (Goldstein et al., 2001)

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial\mathcal{L}}{\partial\dot{q}}(q,\dot{q})\right) - \frac{\partial\mathcal{L}}{\partial q}(q,\dot{q}) = \tau \tag{5.24}$$

where $q \in \mathbb{R}^n$ is a vector of generalised coordinates and $\tau \in \mathbb{R}^n$ is the vector of generalised forces. We note that the position variables are written as $q \in \mathbb{R}^n$ and the velocity variables are written as $v = \dot{q} \in \mathbb{R}^n$. This is a convenient choice of state variables for many systems, but as we will see later, there are also many systems for which the position and velocity variables cannot be written in this form. The Lagrangian is given by

$$\mathcal{L}(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} := \mathcal{U}(q, \dot{q}) - \mathcal{V}(q). \tag{5.25}$$

Here, $\mathcal{U}(q, \dot{q})$ is the kinetic and $\mathcal{V}(q)$ the potential energy functions. We assume that the kinetic energy function is positive definite and in the form

$$\mathcal{U}(q, \dot{q}) := \frac{1}{2} \dot{q}^{\mathsf{T}} M(q) \dot{q}, \tag{5.26}$$

where $M(q)$ is inertia matrix. For a kinetic energy function written in this form we can recast the Euler-Lagrange equations (5.25) into the equivalent form

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + n(q) = \tau \tag{5.27}$$

where $C(q, \dot{q})$ is the Coriolis and centripetal matrix and $n(q)$ is the vector of potential forces defined as

$$n(q) := \frac{\partial V}{\partial q}(q). \tag{5.28}$$

We can also rewrite (5.27) in workspace coordinates $x_{0e}^0 \in \mathbb{R}^6$. The relation between the joint and operational space velocities is then given by the Jacobian

$$\dot{x}_{0e} = J(q)\dot{q}, \qquad J(q) = \frac{\partial f}{\partial q} \tag{5.29}$$

where f is the mapping $f : q \to x_{0e}^0$. If this mapping is smooth and invertible we can write

$$\dot{q} = J^{-1}(q)\dot{x}_{0e}, \tag{5.30}$$

$$\ddot{q} = J^{-1}(q)\ddot{x}_{0e} + \frac{\mathrm{d}}{\mathrm{d}t}(J^{-1}(q))\dot{x}_{0e}. \tag{5.31}$$

We can now substitute this into (5.27) and pre-multiply by $J^{-\mathsf{T}}$ which gives

$$J^{-\mathsf{T}} M J^{-1} \ddot{x}_{0e} + \left(J^{-\mathsf{T}} C J^{-1} + J^{-\mathsf{T}} M \frac{\mathrm{d}}{\mathrm{d}t}(J^{-1}) \right) \dot{x}_{0e} + J^{-\mathsf{T}} n = J^{-\mathsf{T}} \tau. \tag{5.32}$$

This can be written as

$$\bar{M}(q)\ddot{x}_{0e} + \bar{C}(q, \dot{q})\dot{x}_{0e} + \bar{n}(q) = \bar{\tau} \tag{5.33}$$

where

$$\bar{M}(q) = J^{-\mathsf{T}}(q) M(q) J^{-1}(q), \tag{5.34}$$

$$\bar{C}(q) = J^{-\mathsf{T}}(q) \left(C(q, \dot{q}) J^{-1}(q) + M \frac{\mathrm{d}}{\mathrm{d}t}(J^{-1}(q)) \right), \tag{5.35}$$

$$\bar{n}(q) = J^{-\mathsf{T}}(q) n(q), \tag{5.36}$$

$$\bar{\tau}(q) = J^{-\mathsf{T}}(q) \tau. \tag{5.37}$$

5.5.1 The Boundedness Property

Ghorbel et al. (1998) identify all manipulators for which Property 1, i.e. the boundedness condition of the inertia matrix, is satisfied. They find that for a large class of manipulators, including manipulators with only revolute or only prismatic joints, this property is always satisfied. They do not address the mathematical representation of the inertia matrix. A bad choice of state variables may result in an unbounded inertia matrix due to mathematical singularities. However, as long as we can use generalised coordinates to represent the state of the robot, as in (5.27), we can do this without the presence of singularities. In this case, all manipulators characterised by Ghorbel et al. (1998) to satisfy Property 1 from a design point of view, will also satisfy Property 1 from mathematical point of view, as is our main concern.

This is also true for the formulation in (5.33). However, we note that $J(q)$ is not always invertible. This is not due to singularities in the mathematical representation, but due to kinematic singularities. Kinematic singularities are configurations at which the mobility of the manipulator is reduced, i.e. it is not possible to impose an arbitrary motion to the end effector. This is thus a property of the manipulator design and not due to singularities in the mathematical representation.

The Jacobian of a robotic manipulator has a very simple form: The columns representing a rotation around an axis z_1 are in the form $J_1 = \begin{bmatrix}(z_1 \times p_1)^\mathsf{T} & z_1^\mathsf{T}\end{bmatrix}^\mathsf{T}$ for a point p_1 on z_1, and the columns representing a translation along the axis z_2 are in the form $J_1 = \begin{bmatrix}z_2^\mathsf{T} & 0\end{bmatrix}^\mathsf{T}$.

Example 5.1. As an example we use the geometric Jacobian of the SCARA robot (Murray et al., 1994) which has three revolute joints and one prismatic joint, all with respect to the same axis z. The end effector of the SCARA manipulator can thus translate freely in $\mathbb{R}^3$ and rotate around the z-axis. This is known as the Schoenflies motion. The Jacobian is given by

$$J(q) = \begin{bmatrix} z \times p_1 & z \times p_2 & z \times p_3 & z \\ z & z & z & 0 \end{bmatrix} \tag{5.38}$$

$$= \begin{bmatrix} 0 & l_1 \cos q_1 & l_1 \cos q_1 + l_2 \cos(q_1 + q_2) & 0 \\ 0 & l_1 \sin q_1 & l_1 \sin q_1 + l_2 \sin(q_1 + q_2) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \tag{5.39}$$

where l_1 is the length of link 1, l_2 is the length of link 2 and $z = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^\mathsf{T}$. The determinant of the 4×4-matrix of (5.39) representing the 4-DoF motion (cancelling rows 4 and 5) is given by

$$\det(J(q)) = l_1 l_2 (\cos q_1 \sin(q_1 + q_2) + \sin q_1 \cos(q_1 + q_2)). \tag{5.40}$$

We note that the determinant is zero only if $q_2 = \{0, \pi/2\}$. This is the case when the arm is stretched out ($q_2 = 0$) and the manipulator loses mobility. This kinematic singularity is thus due to the geometry of the manipulator and not due to the mathematical representation.

We see from (5.38) that manipulators with only revolute and prismatic joints will not have mathematical singularities. We can assume that the configurations that lead to kinematic singularities are avoided and thus that $J(q)$ is invertible for standard industrial manipulators. Since $J(q)$ is invertible Property 1 holds also when the dynamics are written in the form of (5.33).

5.5.2 The Skew-Symmetric Property

For robotic manipulators represented in generalised coordinates, the Coriolis matrix is normally obtained by the Christoffel symbols of the first kind as (Murray et al., 1994)

$$C(q,\dot{q}) := \{c_{ij}\} = \left\{ \sum_{k=1}^{n} c_{ijk}\dot{q}_k \right\}, \tag{5.41}$$

$$c_{ijk} := \frac{1}{2}\left(\frac{\partial m_{ij}}{\partial q_k} + \frac{\partial m_{ik}}{\partial q_j} - \frac{\partial m_{kj}}{\partial q_i} \right) \tag{5.42}$$

where $M(q) = \{m_{ij}\}$. Given this representation it is straight forward to show that the matrix $(\dot{M} - 2C)$ is skew-symmetric (Murray et al., 1994; Sciavicco and Siciliano, 2005). We see this if we write out the components of $(\dot{M} - 2C)$:

$$\begin{aligned}
(\dot{M} - 2C)_{ij} &= \dot{m}_{ij} - 2c_{ij} \\
&= \sum_{k=1}^{n} \left(\frac{\partial m_{ij}}{\partial q_k}\dot{q}_k - \frac{\partial m_{ij}}{\partial q_k}\dot{q}_k - \frac{\partial m_{ik}}{\partial q_j}\dot{q}_k + \frac{\partial m_{kj}}{\partial q_i}\dot{q}_k \right) \\
&= \sum_{k=1}^{n} \left(\frac{\partial m_{kj}}{\partial q_i}\dot{q}_k - \frac{\partial m_{ik}}{\partial q_j}\dot{q}_k \right).
\end{aligned} \tag{5.43}$$

We see that $(\dot{M} - 2C)^{\mathsf{T}} = -(\dot{M} - 2C)$ and Property 2 is satisfied.

We note that to obtain the Coriolis matrix we multiply the position variables and the velocity variables. For example, we multiply $\frac{\partial m_{ij}(q)}{\partial q_k}$ with $\dot{q}_k$ and get

$$\{c_{ij}\} = \frac{1}{2}\left\{ \sum_{k=1}^{n} \left(\frac{\partial m_{ij}}{\partial q_k} + \frac{\partial m_{ik}}{\partial q_j} - \frac{\partial m_{kj}}{\partial q_i} \right)\dot{q}_k \right\}, \tag{5.44}$$

which only makes sense if the derivative of the position variable equals the velocity variable. This is the case for robot manipulators with revolute or prismatic joints, but not for vehicles with configuration space $SO(3)$ or $SE(3)$. We will say that a configuration space is non-Euclidean when the velocity variable cannot be written simply as the time-derivative of the position variable. The Christoffel symbols can thus not be used to derive the Coriolis matrix for vehicle-manipulator systems where the vehicle has a non-Euclidean configuration space.

For the formulation in (5.33) we compute

$$\dot{\bar{M}} - 2\bar{C} = J^{-\mathsf{T}}\dot{M}J^{-1} - 2J^{-\mathsf{T}}M\frac{\mathrm{d}}{\mathrm{d}t}J^{-1} - 2J^{-\mathsf{T}}CJ^{-1} + 2J^{-\mathsf{T}}M\frac{\mathrm{d}}{\mathrm{d}t}(J^{-1}) \tag{5.45}$$

$$= J^{-\mathsf{T}}(\dot{M} - 2C)J^{-1} - 2J^{-\mathsf{T}}M\frac{\mathrm{d}}{\mathrm{d}t}J^{-1} + 2J^{-\mathsf{T}}M\frac{\mathrm{d}}{\mathrm{d}t}(J^{-1}) \tag{5.46}$$

$$= J^{-\mathsf{T}}(\dot{M} - 2C)J^{-1} \tag{5.47}$$

which, given that $(\dot{M} - 2C)$ is skew-symmetric, is skew-symmetric.

5.6 Vehicle Dynamics

The dynamics of a single rigid body, such as an underwater vehicle, are usually given by (Fossen, 2002)

$$\dot{\eta} = J(\eta)\nu, \tag{5.48}$$

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + n(\eta) = \tau \tag{5.49}$$

where $\eta = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^{\mathsf{T}}$ is the position and orientation of the vehicle in the reference frame and $\nu = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^{\mathsf{T}}$ is the linear and angular velocities in the body frame. $D(\nu)$ is the friction and damping matrix present for underwater vehicles and $n(\eta)$ represents the gravitational and buoyancy forces.

The kinematics of the system (5.48) is given by the velocity transformation matrix $J(\eta)$ which gives the relation between the local and global velocity variables. $J(\eta)$ is defined as

$$J(\eta) = \begin{bmatrix} R_{b0}(\Theta) & 0 \\ 0 & T_\Theta(\Theta) \end{bmatrix} \tag{5.50}$$

where $R_{b0}(\Theta)$ is the rotation matrix and depends only on the orientation of the vessel represented by the Euler angles $\Theta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^{\mathsf{T}}$, given in the reference frame. $T_\Theta(\Theta)$ is given by (zyx-sequence)

$$T_\Theta(\Theta) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix}. \tag{5.51}$$

We note that $T_\Theta(\Theta)$, and thus also $J(\eta)$, is not defined for $\theta = \pm\pi/2$.

Similarly to robotic manipulators we can rewrite the dynamics using general coordinates η, eliminating the body frame coordinates ν from the equations. The dynamics are then written as

$$\tilde{M}(\eta)\ddot{\eta} + \tilde{C}(\eta,\dot{\eta})\dot{\eta} + \tilde{D}(\eta,\dot{\eta})\dot{\eta} + \tilde{n}(\eta) = \tilde{\tau} \tag{5.52}$$

where

$$\tilde{M}(\eta) = J^{-\mathsf{T}}(\eta) M J^{-1}(\eta) \tag{5.53}$$

$$\tilde{n}(\eta) = J^{-\mathsf{T}}(\eta) n(\eta) \tag{5.54}$$

$$\tilde{\tau} = J^{-\mathsf{T}}(\eta) \tau \tag{5.55}$$

$$\tilde{D}(\eta, \dot{\eta}) = J^{-\mathsf{T}}(\eta) D(J^{-1}(\eta)\dot{\eta}) J^{-1}(\eta) \tag{5.56}$$

$$\tilde{C}(\eta, \dot{\eta}) = J^{-\mathsf{T}}(\eta) \left[C(J^{-1}(\eta)\eta) - M J^{-1}(\eta) \dot{J}(\eta) \right] J^{-1}(\eta) \tag{5.57}$$

5.6.1 The Boundedness Property

First note that for the system (5.48-5.49) the inertia matrix is always bounded as it is independent of the position variables. In this sense it is advantageous to formulate the dynamics in the body frame. Consider the system (5.52-5.57) and recall that $J(\eta)$ is not defined for $\theta = \pm\pi/2$. This is the well known Euler angle singularity for the zyx-sequence. The inverse mappings $T_{\Theta}^{-1}(\Theta)$ and $J^{-1}(\eta)$ are defined for all $\theta \in \mathbb{R}$ but singular for $\theta = \pm\pi/2$. The boundedness of the inertia matrix in Property 1 is thus not true. We can only obtain a weaker result than the one found in Property 1:

Property 3. (The weak boundedness property) The inertia matrix $\tilde{M}(\eta)$ is uniformly bounded in η for θ separated from $\pm\pi/2$, i.e. there exists constants d_1 and d_2 such that

$$0 < d_1 \leq \left|\left| \tilde{M}(\eta) \right|\right| \leq d_2 < \infty, \quad \forall\, \eta \in \mathbb{R}^6 \backslash \{ \left| |\theta| - \frac{\pi}{2} \right| \geq \delta \} \tag{5.58}$$

for some small positive delta. We note that the lower bound $d_1 > 0$ only exists if $|\theta|$ is separated from $\pi/2$ by some constant δ.

When $\theta = \pi/2$ we have $d_1 = 0$ and (5.3) does not hold as $d = 1$.

This singularity can be avoided by using the unit quaternion representation, which does not have a singularity at the cost of introducing a fourth parameter to describe the orientation. However, in computing the Euler angles from the quaternions the Euler angle singularity reappears.

5.6.2 The Skew-Symmetric Property

There are many ways to choose the Coriolis matrix so that $\dot{M} - 2C$ is skew-symmetric. We first note that if M is constant, and thus $\dot{M} = 0$, this is true if C is skew-symmetric. The Coriolis matrix can for example be written as

$$C(\nu) = \begin{bmatrix} 0 & \widehat{M_{11}\nu_1} + \widehat{M_{12}\nu_2} \\ \widehat{M_{11}\nu_1} + \widehat{M_{12}\nu_2} & \widehat{M_{21}\nu_1} + \widehat{M_{22}\nu_2} \end{bmatrix} \tag{5.59}$$

where $\nu = \begin{bmatrix} \nu_1^{\mathsf{T}} & \nu_2^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$ and $\hat{\lambda}$ is the skew-symmetric matrix representation of $\lambda \in \mathbb{R}^3$ given by

$$\hat{\lambda} = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix} \in so(3). \tag{5.60}$$

Alternatively, from Kirchhoff's equations we get

$$C(\nu) = -\begin{bmatrix} 0 & \widehat{\frac{\partial \mathcal{U}}{\partial \nu_1}} \\ \widehat{\frac{\partial \mathcal{U}}{\partial \nu_1}} & \widehat{\frac{\partial \mathcal{U}}{\partial \nu_2}} \end{bmatrix} \tag{5.61}$$

where $\mathcal{U}(\eta)$ is given as in (5.26). The proof is found in Sagatun and Fossen (1992). Both these representations satisfy Property 2. Several other representations for which the Coriolis matrix is skew-symmetric are found in Fossen and Fjellstad (1995).

If the inertia matrix is not constant and the dynamics are in the form of (5.52), we can also show this property. The time derivative of the inertia matrix can be written as (Fossen, 1991)

$$\dot{\tilde{M}} = J^{-\mathsf{T}}(\dot{M} - 2MJ^{-1}\dot{J})J^{-1} \tag{5.62}$$

and we get

$$\begin{aligned}(\dot{\tilde{M}} - 2\tilde{C}) =& J^{-\mathsf{T}}\dot{M}J^{-1} - 2J^{-\mathsf{T}}MJ^{-1}\dot{J}J^{-1} - 2J^{-\mathsf{T}}CJ^{-1} + 2J^{-\mathsf{T}}MJ^{-1}\dot{J}J^{-1} \\ =& J^{-\mathsf{T}}(\dot{M} - 2C)J^{-1}\end{aligned} \tag{5.63}$$

and thus, as $(\dot{M} - 2C)$ is skew-symmetric, so is $(\dot{\tilde{M}} - 2\tilde{C})$ and $(\dot{\bar{M}} - 2\bar{C})$.

5.7 Multibody Dynamics with a Free-Floating Base

In this section we review some commonly used approaches for modelling multibody systems and propose a new approach that has certain advantages when it comes to the boundedness and skew-symmetric properties addressed in this paper. Consider the setup of Fig. 5.1 describing a general n-link robot manipulator arm attached to a free-moving base. Choose an inertial coordinate frame Ψ_0, a frame Ψ_b rigidly attached to the moving base, and n frames Ψ_i (not shown) attached to each link i at the centre of mass with axes aligned with the principal directions of inertia. Finally, choose a vector $q \in \mathbb{R}^n$ that describes the configuration of the n joints.

5.7.1 The Model of Schjølberg (1996)

In this section we present the dynamic equations as they are normally presented in the underwater robotics literature. The details can be found in Schjølberg (1996). The dynamics can be written as

$$\dot{\xi} = J(\xi)\zeta, \tag{5.64}$$

$$M(q)\dot{\zeta} + C(q,\zeta)\zeta + D(q,\zeta)\zeta + n(\xi) = \tau \tag{5.65}$$

where $\xi = \begin{bmatrix}\eta^{\mathsf{T}} & q^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$, $\zeta = \begin{bmatrix}\nu^{\mathsf{T}} & \dot{q}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$, $M(q) \in \mathbb{R}^{(6+n)\times(6+n)}$ is the inertia matrix, $C(q,\zeta) \in \mathbb{R}^{(6+n)\times(6+n)}$ is the Coriolis and centripetal matrix and $D(\xi,\zeta) \in \mathbb{R}^{(6+n)\times(6+n)}$ is the damping matrix. The velocity transformation matrix is given by

$$J(\xi) = \begin{bmatrix} R_{0b}(\Theta) & 0 & 0 \\ 0 & T_\Theta(\Theta) & 0 \\ 0 & 0 & I \end{bmatrix} \in \mathbb{R}^{(6+n)\times(6+n)}. \tag{5.66}$$

where I (no subscript) denotes the identity matrix. Similarly to the dynamics of the vehicle with no robotic arm, the vehicle-manipulator equations can be written in the form of Equations (5.53-5.57) but with the velocity transformation matrix as in (5.66). Let ν_{0i}^0 denote the linear and angular velocity of body i represented in the inertial frame, and $P_i(q) \in \mathbb{R}^{6\times(6+n)}$ be the transformation matrix of link i, that gives the relation

$$\nu_{0i}^0 = P_i(q)\zeta. \tag{5.67}$$

The inertia matrix of the vehicle-manipulator system can then be written as (Egeland and Pettersen, 1998)

$$M(q) = \sum_{i=b}^{n} P_i^\mathsf{T}(q) I_i P_i(q) \tag{5.68}$$

where $I_i \in \mathbb{R}^{6\times6}$ denotes the constant positive-definite diagonal inertia tensor of link i expressed in Ψ_i and we thus sum from the base b to the end of the chain, i.e., link n. We note that the inertia matrix is independent of the position η of the vehicle.

The Coriolis matrix is given by

$$C(q,\zeta) = \sum_{i=b}^{n} \dot{P}_i^\mathsf{T}(q) I_i P_i(q) - P_i^\mathsf{T}(q) W_i(\zeta) P_i(q) \tag{5.69}$$

where $W_i(\zeta)$ is a skew-symmetric matrix (Schjølberg, 1996). We will use the framework of Egeland and Pettersen (1998) to find an expression for $W_i(\zeta)$. This is shown in Section 5.7.2.

Alternatively we can write the dynamics in terms of the vector $\varsigma = \begin{bmatrix}\eta^\mathsf{T} & (x_{0e}^0)^\mathsf{T}\end{bmatrix}^\mathsf{T} \in \mathbb{R}^{12}$ where x_{0e}^0 is the manipulator position/orientation vector in the inertial frame. We present the equations as first presented in Schjølberg (1996), but correct an error in the expression of the Coriolis matrix. The dynamics can be written as

$$\bar{M}(\eta,q)\ddot{\varsigma} + \bar{C}(\eta,q,\zeta)\dot{\varsigma} + \bar{D}(\eta,q,\zeta)\dot{\varsigma} + \bar{n}(\eta,q) = \bar{\tau} \tag{5.70}$$

with the matrices as in (5.53-5.57), but with

$$\bar{J}(\eta) = \begin{bmatrix} J(\eta) & 0 \\ \bar{R}_{b0}(\Theta)\bar{J}_1(\eta) & \bar{R}_{b0}(\Theta)\bar{J}_2(q) \end{bmatrix} \in \mathbb{R}^{12\times(6+n)} \tag{5.71}$$

where $J(\eta)$ is given by (5.50) and $\bar{J}_1$ and $\bar{J}_2$ satisfy

$$\dot{x}_{be}^b = \bar{J}_1(\eta)\nu + \bar{J}_2(q)\dot{q} \tag{5.72}$$

where x_{be}^b is the end-effector position/orientation in the base frame and

$$\bar{R}_{b0}(\Theta) = \begin{bmatrix} R_{b0} & 0 \\ 0 & R_{b0} \end{bmatrix}. \tag{5.73}$$

This formulation is convenient because it allows us to write the dynamics in terms of the end-effector position and orientation directly.

5.7.2 Multibody Dynamics in Terms of Quasi-Velocities

In this section we derive the dynamics of a robotic manipulator mounted on a free-floating base in terms of quasi-velocities. The approach is based on Egeland and Pettersen (1998), but a few errors from this paper have been corrected and we also provide some more details in the derivation.

First, write the linear and angular velocities ν_{0i}^0 of each link i represented in the inertial frame (frame 0) as

$$\nu_{0i}^0 = \begin{bmatrix} \nu_{0i,v}^0 \\ \nu_{0i,\omega}^0 \end{bmatrix} = \frac{\partial \nu_{0i}^0}{\partial \zeta} \zeta. \tag{5.74}$$

Then the dynamics can be written as (Egeland and Pettersen, 1998)

$$\sum_{i=b}^{6} \left\{ \frac{\partial \nu_{0i}^0}{\partial \zeta}^{\mathsf{T}} \left[\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial \mathcal{U}_i}{\partial \nu_{0i}^0}^{\mathsf{T}} + \begin{bmatrix} \hat{\nu}_{0i,\omega}^0 & 0 \\ \hat{\nu}_{0i,v}^0 & \hat{\nu}_{0i,\omega}^0 \end{bmatrix} \frac{\partial \mathcal{U}_i}{\partial \nu_{0i}^0}^{\mathsf{T}} \right] \right\} = \tau \tag{5.75}$$

where

$$\tau = \sum_{i=b}^{6} \left\{ \frac{\partial \nu_{0i}^0}{\partial \zeta}^{\mathsf{T}} \right\}. \tag{5.76}$$

We now derive the dynamics in matrix form following the approach in Egeland and Pettersen (1998), but in addition we show the explicit expressions for the matrices which were not shown in Egeland and Pettersen (1998) and we correct an error is the expression of the Coriolis matrix. First write

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial \mathcal{U}_i}{\partial \nu_{0i}^0} = \frac{\mathrm{d}}{\mathrm{d}t} (I_i \nu_{0i}^0) = I_i \dot{\nu}_{0i}^0 = I_i \left(\frac{\partial \nu_{0i}^0}{\partial \zeta} \dot{\zeta} + \frac{\partial \dot{\nu}_{0i}^0}{\partial \zeta} \zeta \right), \tag{5.77}$$

and

$$\begin{aligned} \begin{bmatrix} \hat{\nu}_{0i,\omega}^0 & 0 \\ \hat{\nu}_{0i,v}^0 & \hat{\nu}_{0i,\omega}^0 \end{bmatrix} \begin{bmatrix} \frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^0} \\ \frac{\partial \mathcal{U}_i}{\partial \nu_{0i,\omega}^0} \end{bmatrix} &= \begin{bmatrix} \hat{\nu}_{0i,\omega}^0 \frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^0} & 0 \\ \hat{\nu}_{0i,v}^0 \frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^0} & \hat{\nu}_{0i,\omega}^0 \frac{\partial \mathcal{U}_i}{\partial \nu_{0i,\omega}^0} \end{bmatrix} \\ &= \begin{bmatrix} -\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^0} \times \nu_{0i,\omega}^0 & 0 \\ -\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^0} \times \nu_{0i,v}^0 & -\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,\omega}^0} \times \nu_{0i,\omega}^0 \end{bmatrix} \\ &= - \begin{bmatrix} 0 & \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^0}} \\ \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^0}} & \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,\omega}^0}} \end{bmatrix} \begin{bmatrix} \nu_{0i,v}^0 \\ \nu_{0i,\omega}^0 \end{bmatrix}. \end{aligned} \tag{5.78}$$

Substituting (5.77) and (5.78) into (5.75) we get

$$\begin{aligned}
\sum_{i=b}^{6}\left\{\frac{\partial \nu_{0i}^{0}}{\partial \zeta}^{\mathsf{T}}\left[\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{U}_i}{\partial \nu_{0i}^{0}}^{\mathsf{T}}+\begin{bmatrix}\hat{\nu}_{0i,\omega}^{0} & 0\\ \hat{\nu}_{0i,v}^{0} & \hat{\nu}_{0i,\omega}^{0}\end{bmatrix}\frac{\partial \mathcal{U}_i}{\partial \nu_{0i}^{0}}^{\mathsf{T}}\right]\right\} &= \tau \\
\sum_{i=b}^{6}\left\{\frac{\partial \nu_{0i}^{0}}{\partial \zeta}^{\mathsf{T}}\left[I_i\left(\frac{\partial \nu_{0i}^{0}}{\partial \zeta}\dot{\zeta}+\frac{\dot{\partial \nu_{0i}^{0}}}{\partial \zeta}\zeta\right)-\begin{bmatrix}0 & \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^{0}}}\\ \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^{0}}} & \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,\omega}^{0}}}\end{bmatrix}\begin{bmatrix}\nu_{0i,v}^{0}\\ \nu_{0i,\omega}^{0}\end{bmatrix}\right]\right\} &= \tau \\
\sum_{i=b}^{6}\left\{\frac{\partial \nu_{0i}^{0}}{\partial \zeta}^{\mathsf{T}} I_i\frac{\partial \nu_{0i}^{0}}{\partial \zeta}\dot{\zeta}+\frac{\partial \nu_{0i}^{0}}{\partial \zeta}^{\mathsf{T}} I_i\frac{\dot{\partial \nu_{0i}^{0}}}{\partial \zeta}\zeta-\frac{\partial \nu_{0i}^{0}}{\partial \zeta}^{\mathsf{T}}\begin{bmatrix}0 & \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^{0}}}\\ \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^{0}}} & \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,\omega}^{0}}}\end{bmatrix}\begin{bmatrix}\nu_{0i,v}^{0}\\ \nu_{0i,\omega}^{0}\end{bmatrix}\right\} &= \tau \\
\sum_{i=b}^{6}\left[\frac{\partial \nu_{0i}^{0}}{\partial \zeta}^{\mathsf{T}} I_i\frac{\partial \nu_{0i}^{0}}{\partial \zeta}\right]\dot{\zeta}+\sum_{i=b}^{6}\left[\frac{\partial \nu_{0i}^{0}}{\partial \zeta}^{\mathsf{T}} I_i\frac{\dot{\partial \nu_{0i}^{0}}}{\partial \zeta}-\frac{\partial \nu_{0i}^{0}}{\partial \zeta}^{\mathsf{T}}\begin{bmatrix}0 & \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^{0}}}\\ \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^{0}}} & \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,\omega}^{0}}}\end{bmatrix}\frac{\partial \nu_{0i}^{0}}{\partial \zeta}\right]\zeta &= \tau
\end{aligned} \tag{5.79}$$

where we have used the relation in (5.74). The inertia and Coriolis matrices are then found from (5.79) as

$$M(q)=\sum_{i=b}^{n}P_i^{\mathsf{T}}(q)I_iP_i(q) \tag{5.80}$$

where

$$P_i(q)=\frac{\partial \nu_{0i}^{0}}{\partial \zeta} \tag{5.81}$$

and

$$C(q,\zeta)=\sum_{i=b}^{n}\left(P_i^{\mathsf{T}}(q)I_i\dot{P}_i(q)-P_i^{\mathsf{T}}(q)W_iP_i(q)\right) \tag{5.82}$$

where

$$W_i(\nu_{0i}^{0})=\begin{bmatrix}0 & \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^{0}}}\\ \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,v}^{0}}} & \widehat{\frac{\partial \mathcal{U}_i}{\partial \nu_{0i,\omega}^{0}}}\end{bmatrix}. \tag{5.83}$$

5.7.3 General Multibody Dynamics

In this section we extend the formulation in the previous section to include more general structures and also mechanisms where the position of the vehicle needs to be included in the dynamics. The approach is based on Duindam and Stramigioli (2008) and From et al. (2009a) where the dynamics of vehicle-manipulator systems are derived and the boundedness property holds. However, for the Coriolis matrix presented in From et al. (2009a) the skew-symmetric property does not hold. In this section we thus present a new formulation of the dynamic equations for vehicle-manipulator systems where both the boundedness and the skew-symmetric properties hold and which allows us to include the vehicle configuration in the representation.

Using standard notation (Murray et al., 1994), we can describe the pose of each frame Ψ_i relative to Ψ_0 as a homogeneous transformation matrix $g_{0i} \in SE(3)$ of the form

$$g_{0i} = \begin{bmatrix} R_{0i} & p_{0i} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{5.84}$$

with rotation matrix $R_{0i} \in SO(3)$ and translation vector $p_{0i} \in \mathbb{R}^3$. This pose can also be described using the vector of joint coordinates q as

$$g_{0i} = g_{0b} g_{bi} = g_{0b} g_{bi}(q). \tag{5.85}$$

The base pose g_{0b} and the joint positions q thus fully determine the configuration state of the robot. In a similar way, the spatial velocity of each link can be expressed using twists (Murray et al., 1994):

$$\nu_{0i}^0 = \begin{bmatrix} \nu_{0i,v}^0 \\ \nu_{0i,\omega}^0 \end{bmatrix} = \nu_{0b}^0 + \nu_{bi}^0 = \mathrm{Ad}_{g_{0b}} \left(\nu_{0b}^b + J_i(q)\dot{q} \right) \tag{5.86}$$

where $\nu_{0i,v}^0$ and $\nu_{0i,\omega}^0$ are the linear and angular velocities, respectively, of link i relative to the inertial frame, $J_i(q) \in \mathbb{R}^{6\times n}$ is the geometric Jacobian of link i relative to Ψ_b and the adjoint is defined as $\mathrm{Ad}_g := \left[\begin{smallmatrix} R & \hat{p}R \\ 0 & R \end{smallmatrix} \right] \in \mathbb{R}^{6\times 6}$. The velocity state is thus fully determined given the twist ν_{0b}^b of the base and the joint velocities $\dot{q}$.

This illustrates how the kinematics of the system can be naturally described in terms of the (global) state variables $Q = \{g_{0b}, q\}$ and $v = \{\nu_{0b}^b, \dot{q}\}$. We will use these observations to reformulate the relation given in (5.74). We will also rewrite the inertia matrix, as given in (5.26) and (5.80), in terms of the globally defined state variables.

Given a mechanism with coordinates formulated in this generalised form, we can write its kinetic energy as $\mathcal{U}(Q, v) = \frac{1}{2} v^\mathsf{T} M(Q) v$ with $M(Q)$ the inertia matrix in coordinates Q. The dynamics of this system then satisfy

$$M(Q)\dot{v} + C(Q, v)v = \tau \tag{5.87}$$

with τ the vector of gravitational forces, friction, and other external forces (collocated with v).

From expression (5.86) for the twist of each link in the mechanism, we can derive an expression for the total kinetic energy. The kinetic energy $\mathcal{U}_i$ of link i then follows as

$$\begin{aligned} \mathcal{U}_i &= \frac{1}{2} \left(\nu_{0i}^0 \right)^\mathsf{T} I_i \nu_{0i}^0 \\ &= \frac{1}{2} \left(\nu_{0b}^b + J_i(q)\dot{q} \right)^\mathsf{T} \mathrm{Ad}_{g_{ib}}^\mathsf{T} I_i \, \mathrm{Ad}_{g_{ib}} \left(\nu_{0b}^b + J_i(q)\dot{q} \right) \\ &= \frac{1}{2} \begin{bmatrix} \left(\nu_{0b}^b \right)^\mathsf{T} & \dot{q}^\mathsf{T} \end{bmatrix} M_i(q) \begin{bmatrix} \nu_{0b}^b \\ \dot{q} \end{bmatrix} \\ &= \frac{1}{2} v^\mathsf{T} M_i(q) v \end{aligned} \tag{5.88}$$

with

$$M_i(q) := \begin{bmatrix} \mathrm{Ad}_{g_{ib}}^\mathsf{T} I_i \, \mathrm{Ad}_{g_{ib}} & \mathrm{Ad}_{g_{ib}}^\mathsf{T} I_i \, \mathrm{Ad}_{g_{ib}} J_i \\ J_i^\mathsf{T} \, \mathrm{Ad}_{g_{ib}}^\mathsf{T} I_i \, \mathrm{Ad}_{g_{ib}} & J_i^\mathsf{T} \, \mathrm{Ad}_{g_{ib}}^\mathsf{T} I_i \, \mathrm{Ad}_{g_{ib}} J_i \end{bmatrix} \tag{5.89}$$

where $J_i(q)$ is the geometric Jacobian of link i. The total kinetic energy of the mechanism is given by the sum of the kinetic energies of the mechanism links and the non-inertial base, that is,

$$\mathcal{U}(q,v) = \frac{1}{2} v^{\mathsf{T}} \underbrace{\left(\begin{bmatrix} I_b & 0 \\ 0 & 0 \end{bmatrix} + \sum_{i=1}^{n} M_i(q) \right)}_{\text{inertia matrix } M(q)} v \tag{5.90}$$

with $M(q)$ the inertia matrix of the total system. Note that neither $\mathcal{U}(q,v)$ nor $M(q)$ depend on the pose g_{0b} and hence the choice of inertial reference frame Ψ_0.

We see that from (5.89) we can reformulate the expression in Egeland and Pettersen (1998) for the inertia matrix and we get

$$M(q) = \sum_{i=b}^{n} P_i^{\mathsf{T}}(q) I_i P_i(q) \tag{5.91}$$

where

$$P_i(q) = \begin{bmatrix} \mathrm{Ad}_{g_{ib}} & \mathrm{Ad}_{g_{ib}} J_i \end{bmatrix} \in \mathbb{R}^{6\times(6+n)} \tag{5.92}$$

and the Jacobian J_i of link i is given by

$$J_i(q) = \begin{bmatrix} X_1 & \mathrm{Ad}_{g_{b1}} X_2 & \mathrm{Ad}_{g_{b2}} X_3 & \cdots & \mathrm{Ad}_{g_{b(i-1)}} X_i & 0_{(n-i)\times 6} \end{bmatrix}. \tag{5.93}$$

The partial derivatives of the adjoint map is found by (From et al., 2009a)

$$\frac{\partial \mathrm{Ad}_{g_{ij}}}{\partial q_k} = \begin{cases} \mathrm{Ad}_{g_{i(k-1)}} \mathrm{ad}_{X_k} \mathrm{Ad}_{g_{(k-1)j}} & \text{for } i < k \leq j \\ -\mathrm{Ad}_{g_{i(k-1)}} \mathrm{ad}_{X_k} \mathrm{Ad}_{g_{(k-1)j}} & \text{for } j < k \leq i \\ 0 & \text{otherwise} \end{cases}$$

which also gives us the partial derivative of the Jacobian as

$$\frac{\partial J_i}{\partial q_k} = \begin{bmatrix} 0_{k\times 6} & \frac{\partial \mathrm{Ad}_{g_{bk}}}{\partial q_k} X_{k+1} & \cdots & \frac{\partial \mathrm{Ad}_{g_{b(i-1)}}}{\partial q_k} X_i & 0_{(n-i)\times 6} \end{bmatrix} \tag{5.94}$$

Similarly the Coriolis matrix can be found by

$$C(q,\zeta) = \sum_{i=b}^{n} \left(P_i^{\mathsf{T}}(q) I_i \dot{P}_i(q) - P_i^{\mathsf{T}}(q) W_i P_i(q) \right) \tag{5.95}$$

where W_i is given by (5.83) and P_i by (5.92). $C(q,\zeta)$ is thus also well-defined.

We see that this approach allows us to include forces such as gravity and buoyancy forces. Let the wrench associated with the gravitational force of link i with respect to coordinate frame Ψ_i be given by

$$F_g^i = \begin{bmatrix} f_g \\ \hat{r}_g^0 f_g \end{bmatrix} = -m_{0i} g \begin{bmatrix} R_{0i} e_z \\ \hat{r}_g^0 R_{0i} e_z \end{bmatrix} \tag{5.96}$$

where $e_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\mathsf{T}$ and r_g^0 is the center of mass of link i expressed in frame Ψ_i. In our case Ψ_i is chosen so that r_g^0 coincides with the origin of Ψ_i so we have $r_g^0 = 0$. The equivalent joint torque associated with link i is given by

$$\tau_g^0 = J_i(q)\,\mathrm{Ad}_{g_{0i}}^\mathsf{T}(Q) F_g^i(Q) \tag{5.97}$$

where J_i is the geometric Jacobian and $\mathrm{Ad}_{g_{0i}} = \mathrm{Ad}_{g_{0b}}\,\mathrm{Ad}_{g_{bi}}$ is the transformation from the inertial frame to frame i. We note that both R_{0i} and $\mathrm{Ad}_{g_{0i}}$ depend on the base configuration with respect to the inertial frame. The total effect of the gravity from all the links is then given by $n(Q) = \sum_{i=1}^{n} \tau_g^0$ which enters Equation (5.87) in the same way as the control torque.

We note that to obtain the complete representation of the dynamics we need to make sure we do not leave the manifold when we perform the integration. This can be done either by projecting g_{0b} onto the allowed configuration space $SE(3)$ (McLachlan and Quispel, 2006) or by using structure-preserving integration methods (Munthe-Kaas, 1998).

5.7.4 Multibody Dynamics in Terms of Quasi-Coordinates

We can also follow the generalised Lagrangian method introduced by Duindam and Stramigioli (2008, 2007) and From et al. (2009a). This method gives the dynamic equations for a general mechanism described by a set $Q = \{Q_i\}$ of configuration states Q_i (not necessarily Euclidean), a vector v of velocity states $v_i \in \mathbb{R}^{n_i}$, and several mappings that describe the local Euclidean structure of the configuration states and their relation to the velocity states. More precisely, the neighbourhood of every state $\bar{Q}_i$ is locally described by a set of Euclidean coordinates $\phi_i \in \mathbb{R}^{n_i}$ as $Q_i = Q_i(\bar{Q}_i, \phi_i)$ with $Q_i(\bar{Q}_i, 0) = \bar{Q}_i$, and there exist differentiable matrices S_i such that we can write $v_i = S_i(Q_i, \phi_i)\dot{\phi}_i$ for every Q_i.

The inertia matrix $M(q)$ is then given by (5.90) and the Coriolis matrix $C(Q, v)$ is given by

$$\begin{aligned} \bar{C}_{ij}(Q, v) := &\sum_{k,l} \left(\frac{\partial M_{ij}}{\partial \phi_k} S_{kl}^{-1} - \frac{1}{2} S_{ki}^{-1} \frac{\partial M_{jl}}{\partial \phi_k} \right) \Bigg|_{\phi=0} v_l \\ &+ \sum_{k,l,m,s} \left(S_{mi}^{-1} \left(\frac{\partial S_{mj}}{\partial \phi_s} - \frac{\partial S_{ms}}{\partial \phi_j} \right) S_{sk}^{-1} M_{kl} \right) \Bigg|_{\phi=0} v_l \end{aligned} \tag{5.98}$$

More details and proofs can be found in Duindam and Stramigioli (2008, 2007).

5.7.5 The Boundedness Property

The dynamics as presented in Schjølberg (1996) and Section 5.7.1 do not satisfy Property 1. Due to the singularity there exist isolated points in the configuration space where the inertia matrix is singular. Even though this is the most common formulation of vehicle-manipulator systems in the literature this fact is normally not addressed in Lyapunov stability proofs. The formulation in Egeland and Pettersen (1998) and Section 5.7.2 is globally valid and the inertia matrix is bounded the whole configuration space. For systems where the configuration of non-Euclidean joints needs to be included in the dynamics, there does not seem to be a straight forward way to include the transformation between the local and global state variables without introducing singularities to the formulation.

This is, however, possible with the formulation presented in Sections 5.7.3 and 5.7.4 where the inertia matrix is bounded for the whole configuration space also for non-Eucliden joints with a Lie group topology, such as $SO(3)$ and $SE(3)$. These formulations allow us to use the matrix representation $g_{0i} \in SE(3)$ of the configuration space and the structure of the configuration manifold is thus maintained.

5.7.6 The Skew-Symmetric Property

Schjølberg (1996) show that for the formulation presented in Section 5.7.1 the skew-symmetric property holds in body-fixed coordinates. Based on this proof we can show that this property also holds for the approaches presented in Sections 5.7.2 and 5.7.3. First, for the Coriolis matrix given in (5.69) and (5.82) we can write

$$\begin{aligned}(\dot{M} - 2C) =& \frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_{i=b}^{n} P_i^{\mathsf{T}}(q) I_i P_i(q)\right) - 2\sum_{i=b}^{n}\left(P_i^{\mathsf{T}}(q) I_i \dot{P}_i(q) - P_i^{\mathsf{T}}(q) W_i P_i(q)\right) \\ =& \sum_{i=b}^{n}\left(\dot{P}_i^{\mathsf{T}}(q) I_i P_i(q) + P_i^{\mathsf{T}}(q) I_i \dot{P}_i(q) - 2P_i^{\mathsf{T}}(q) I_i \dot{P}_i(q) + 2P_i^{\mathsf{T}}(q) W_i P_i(q)\right) \\ =& 2\sum_{i=b}^{n} P_i^{\mathsf{T}}(q) W_i P_i(q) \end{aligned} \tag{5.99}$$

and $(\dot{M} - 2C)$ is skew-symmetric, for skew-symmetric W_i. Thus, the formulations given in Sections 5.7.1, 5.7.2 and 5.7.3 all satisfy the skew-symmetric property. This is not true, however, for the parameterisation in 5.7.4.

As we have seen, this property is also true if the system is written in terms of global state variables, as in (5.52) and (5.70), i.e. as long as $(\dot{M} - 2C)$ is skew-symmetric, so is $(\dot{\tilde{M}} - 2\tilde{C})$ and $(\dot{\bar{M}} - 2\bar{C})$.

5.8 Conclusions

The boundedness property of the inertia matrix and the skew-symmetric property of the Coriolis matrix both depend on the choice of mathematical representation. The proofs of such properties thus need to be based on the particular representation chosen. In other words, a reference to a proof for a different choice of state variables or a different parameterisation of the matrices is not valid. In this paper we have shown that several widely used formulations of vehicle-manipulator dynamics do not possess these properties. We have also shown that some of the most commonly used references used in for example in stability proofs of Lyapunov-based control laws in fact do not show these properties. As a result, many of the control laws presented in the literature are not valid.

For several formulations of vehicle-manipulator dynamics commonly found in the literature we have have studied whether the boundedness and skew-symmetric properties hold. When we find the dynamic equations to satisfy these properties we have also included the proofs for future reference. These proofs have not previously been presented correctly for vehicle-manipulator systems. Finally we propose a modified version of the dynamic equations that satisfy both properties for general multibody systems.

Chapter 6

A Singularity Free Formulation of the Dynamically Equivalent Manipulator Mapping for Free-Floating and Free-Flying Space Manipulators

6.1 Abstract

In this paper we derive, for the first time, the singularity-free dynamic equations of the dynamically equivalent manipulator (DEM) of spacecraft-manipulator systems. The DEM is a fixed-base manipulator with the same dynamic properties as the corresponding spacecraft-manipulator system. The DEM consists of a spherical joint, representing the spacecraft, and a robotic arm with the same joint types as the space manipulator. A spherical joint is normally modeled using Euler angles, which leads to singularities, or Euler parameters, which is not a minimal representation and thus not suited for Lagrange's equations. We circumvent these issues by introducing quasi-coordinates which allows us to derive the dynamics using minimal and globally valid non-Euclidean configuration coordinates. This is a great advantage as the configuration space of a spherical joint is non-Euclidean. We thus obtain a computationally efficient and singularity-free formulation of the DEM dynamics with the same complexity as the conventional Lagrangian approach. The closed form formulation makes the proposed approach well suited for system analysis and model-based control. The inertia and Coriolis matrices are presented in such a way that this can be implemented for simulation and control purposes without extensive knowledge of the mathematical background.

Is not included due to copyright

Part II

Robust Manipulator Design

Chapter 7

A Geometric Approach to Handling Torque Failure in Serial and Closed Chain Manipulators with Passive Joints

7.1 Abstract

The increasing use of robotic manipulators in remote and sensitive areas calls for more robust solutions when handling joint failure, and the industry demands mathematically robust approaches to handle even the worst case scenarios. For both serial and parallel manipulators, torque failure is indeed a worst case scenario as the manipulator can collapse due to external forces, such as gravity. It is a very hard task, and in some cases impossible, to prevent damage when this occurs. When possible, however, these aspects should be a concern both in the design of the manipulators and in the control of the manipulator after such an occurrence. Thus, a systematic analysis of the effects of external forces on manipulators with passive joints is presented.

For serial manipulators we investigate under what conditions the robot is conditionally equilibrated, that is, equilibrated with respect to a specific external force. These conditions are, as expected, very restrictive, but for many manipulators we can find certain configurations for which the manipulator does not collapse when the external force is known. The serial, or subchain, case also serves as a good platform for understanding and analysing parallel manipulators. In parallel manipulators passive joints can appear as a design choice or as a result of torque failure. In both cases a good understanding of the effects that passive joints have on the mobility and motion of the parallel manipulator is crucial. We first look at the effect that passive joints have on the mobility of the mechanism. Then, if the mobility, considering pas-

Is not included due to copyright

Part III

Functional Redundancy

Chapter 8

Representing Attitudes as Sets of Frames

8.1 Abstract

A general framework for representing continuous sets of frames with the unit quaternion representation is presented. The determination and control of the attitude of a rigid body is important in a wide range of applications and has been given much attention in the control community. Not always, however, must the desired attitude be restricted to one given orientation, but can be given as a discrete or continuous set of orientations subject to some restriction. An attitude can be represented by the four-parameter unit quaternion without the presence of singularities. It is shown how continuous sets of frames can be described by the unit quaternion representation. It is also shown how this set can be reorientated into an arbitrary coordinate system by the quaternion product. Some work is done on finding the attitude that is closest to the desired orientation when the desired orientation is out of reach due to some restriction on the allowed orientations or rotations.

8.2 Introduction

The attitude control problem of a rigid body is given much attention in the control community, and its applications range from attitude control of aircraft, spacecraft and satellites (Dwyer, 1984; Kristiansen et al., 2005) to rigid bodies held by robotic manipulators (Yuan, 1988; Xian et al., 2004). A thorough discussion on the attitude control problem is given in Wen and Kreutz-Delgado (1991), where global stability is shown for a variety of control laws using the unit quaternion representation in a Lyapunov function.

The unit quaternion group allows orientation and rotation to be represented globally without singularities. One problem of the unit quaternion group is that it is not as easy to visualise as the Euler angles. Many methods have been developed to help visualising quaternions and the relationship between quaternions and three-dimensional rotations. A good introduction on how to visualise quaternions can be found in Hanson (2006) and

Kuipers (2002). Hanson (2006) also gives a thorough presentation of quaternion curves, surfaces and volumes. Of special interest is the work presented on quaternion volumes, where it is shown that a continuous set of frames can be represented by a quaternion and a set of intuitive restrictions in Euler angle representation. The theory of quaternion volumes closely relates to orientation maps. Several techniques that can be used to visualise orientations are discussed in Alpern et al. (1993).

In this paper, the work on quaternion volumes is taken one step further, and a schematic approach on how to represent sets of frames is presented. It is shown how this set can be visualised by a set of points in the unit sphere, and how this set relate to the corresponding quaternion volume. It is also shown how this set can be reorientated so that it is defined with respect to some other reference frame. A test to verify if a quaternion satisfies the restrictions given by the quaternion volume is also presented.

This paper also addresses the problem of how the unit quaternion group can be utilised to find the attitude that is closest to some given orientation when rotations about one axis only are allowed. This work is similar to the results found in Yuan (1988); Wen and Kreutz-Delgado (1991); Hanson (2006) in how the orientation error is presented, but goes one step further in also finding the closest orientation.

8.3 Representing Rotations

Most of the fundamental principles of rotation were presented in two papers by Leonhard Euler in 1775 (Alpern et al., 1993). The first paper shows that any rotation can be accomplished by a sequence of three rotations about the coordinate axes. In the second paper, Euler states that any orientation can be represented by a rotation of some angle ϕ about some fixed axis $\boldsymbol{n}$. He also shows that the composition of two rotations is itself a rotation.

8.3.1 The Unit Quaternion

The unit quaternion representation closely relates to the results presented in Euler's second paper. A good introduction to quaternions is found in Kuipers (2002). Any positive rotation ϕ about a fixed unit vector $\boldsymbol{n}$ can be represented by the four-tuple

$$Q = \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix}, \tag{8.1}$$

where $q_0 \in \mathbb{R}$ is known as the scalar part and $\boldsymbol{q} \in \mathbb{R}^3$ as the vector part. $Q(\phi, \boldsymbol{n})$ is written in terms of ϕ and $\boldsymbol{n}$ by

$$q_0 = \cos\left(\frac{\phi}{2}\right), \qquad \boldsymbol{q} = \sin\left(\frac{\phi}{2}\right)\boldsymbol{n}. \tag{8.2}$$

Q is a quaternion of unit length and denoted a *unit quaternion*. Henceforth, all quaternions have unit length if not other is stated. The quaternion product of a rotation Q followed by a rotation P is written in vector algebra notations as

$$P * Q = \begin{bmatrix} p_0 q_0 - \boldsymbol{p} \cdot \boldsymbol{q} \\ p_0 \boldsymbol{q} + q_0 \boldsymbol{p} + \boldsymbol{p} \times \boldsymbol{q} \end{bmatrix}. \tag{8.3}$$

The cross product implies that quaternion multiplication is not commutative, as expected. Let $P = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix}^\mathsf{T}$ and $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$. Then the quaternion product is written as

$$Q_P * Q = \begin{bmatrix} p_0 q_0 - p_1 q_1 - p_2 q_2 - p_3 q_3 \\ p_0 q_1 + p_1 q_0 + p_2 q_3 - p_3 q_2 \\ p_0 q_2 + p_2 q_0 + p_3 q_1 - p_1 q_3 \\ p_0 q_3 + p_3 q_0 + p_1 q_2 - p_2 q_1 \end{bmatrix}. \tag{8.4}$$

The quaternion product of two unit quaternions is a unit quaternion. By the definition of the quaternion the quaternions Q and $-Q$ produce the same rotation. This is referred to as the dual covering. The quaternion identity is given by $Q_I = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$.

A pure quaternion is a quaternion with zero scalar part. Any vector, $\bar{\boldsymbol{v}} = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T}$ can be represented by a pure quaternion

$$\boldsymbol{v} = \begin{bmatrix} 0 \\ \bar{\boldsymbol{v}} \end{bmatrix}. \tag{8.5}$$

The conjugate of a quaternion is defined as $Q^* = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{bmatrix}^\mathsf{T}$.

8.3.2 Quaternions and Rotations

Let a vector, $\bar{\boldsymbol{v}}_1$, be represented by the pure quaternion $\boldsymbol{v}_1$. This vector can be rotated ϕ radians around the axis $\boldsymbol{n}$ by

$$\boldsymbol{v}_2 = Q * \boldsymbol{v}_1 * Q^*. \tag{8.6}$$

Every vector $\bar{\boldsymbol{v}} \in \mathbb{R}^3$ can be represented by a pure quaternion, hence $\boldsymbol{v}$ is not necessarily a unit quaternion. The quaternion, $Q(\phi, \boldsymbol{n})$, however, is unitary. This represents the angle and the axis that the vector $\bar{\boldsymbol{v}}_1$ is to be rotated about. The resulting vector, $\bar{\boldsymbol{v}}_2$, is then of the same length as $\bar{\boldsymbol{v}}_1$ if and only if Q is a unit quaternion. The quaternion representation also leads to a useful formula for finding the shortest rotation from one orientation to another. Let P and Q be two orientations. Then, by taking

$$E = P^* * Q, \tag{8.7}$$

E will rotate P into Q by the shortest rotation.

Note that equation (8.7) rotates one frame into another frame. By a *frame* it is meant a coordinate system in $\mathbb{R}^3$ using Cartesian coordinates. One frame with respect to another frame represents three degrees of freedom and is referred to as an *attitude orientation*. Equation (8.6) rotates one vector into another vector and has two degrees of freedom (e.g. longitude and latitude) (Ahuactzin and Gupka, 1999). A unit vector with respect to a unit reference vector is referred to as an *attitude direction*. Henceforth, when referred to direction, this is the direction of the z-axis of the body frame with respect to the z-axis of the reference frame.

8.4 Quaternion Volumes

8.4.1 General Definition

A set of frames that correspond to a reference frame by a rotation about a fixed axis, $\boldsymbol{n}$, can be represented by a quaternion and some restriction[1]

$$Q(\phi, \boldsymbol{n}), \qquad \text{for } \phi_{min} < \phi < \phi_{max}. \tag{8.8}$$

When restrictions are not limited to one axis only, a more general description of all allowed orientations can be defined by a combination of rotations given by the quaternion product of two or more quaternions and their restrictions. In this paper, only sets of frames that can be described by a sequence of rotations about fixed axes are treated.

Definition 8.1 (Quaternion Volume)**.** A quaternion volume, $Q^{\otimes}$, is defined as

$$\begin{aligned} Q^{\otimes} \triangleq \{Q(\phi_1, \ldots, \phi_n, \boldsymbol{n}_1, \ldots, \boldsymbol{n}_n) \mid \phi_{1,min} \leq \phi_1 \leq \phi_{1,max} \\ \vdots \\ \phi_{n,min} \leq \phi_n \leq \phi_{n,max}\} \end{aligned} \tag{8.9}$$

for $n \geq 1$ and where

$$Q(\phi_1, \ldots, \phi_n, \boldsymbol{n}_1, \ldots, \boldsymbol{n}_n) = Q(\phi_1, \boldsymbol{n}_1) * \cdots * Q(\phi_n, \boldsymbol{n}_n). \tag{8.10}$$

From the above it is clear that a quaternion volume is obtained by the quaternion product of one or more quaternion volumes. This is stated in the next proposition.

Proposition 8.1 (Quaternion Product of Quat. Volume(s))**.** *The quaternion product of two quaternion volumes, or a quaternion volume and a quaternion, is itself a quaternion volume.*

Proof. By equation (8.3) the quaternion product of two quaternions is a quaternion. Let P be a quaternion with the restrictions $\phi_{min} \leq \phi \leq \phi_{max}$. Then it is a quaternion volume by definition 8.1 with $n = 1$. Then the quaternion product $E = P * Q$ consists of the 16 elements of equation (8.4). Let Q be a quaternion, then E can be written in terms of e_{0-3}.

$$e_0 = p_0(\phi)q_0 - p_1(\phi)q_1 - p_2(\phi)q_2 - p_3(\phi)q_3, \tag{8.11}$$
$$e_1 = p_0(\phi)q_1 + p_1(\phi)q_0 + p_2(\phi)q_3 - p_3(\phi)q_2, \tag{8.12}$$
$$e_2 = p_0(\phi)q_2 + p_2(\phi)q_0 + p_3(\phi)q_1 - p_1(\phi)q_3, \tag{8.13}$$
$$e_3 = p_0(\phi)q_3 + p_3(\phi)q_0 + p_1(\phi)q_2 - p_2(\phi)q_1. \tag{8.14}$$

Note that, as Q is a quaternion, the elements of E are sums of the products of a constant (q_{0-3}) and the elements of the quaternion volume ($p_{0-3}(\phi)$). By representing $P \in \mathbb{R}^4$ and $Q \in \mathbb{R}^4$ as four-tuples, the quaternion product is given by (8.11)-(8.14) and the field

[1] The dual covering allows every rotation to be described twice. In this paper, however, it is only described once, so that all angles are assumed to be in the interval $[-\pi, \pi]$. It is also assumed that all angles of inverse trigonometric functions are in this interval with the correct sign. For $\arctan$, this is denoted $\arctan 2$.

property *closure*[2] is satisfied so that $e_{0-3} \in \mathbb{R}$. Thus, e_{0-3} are functions of ϕ so that the restrictions on ϕ can be applied to the quaternion product. Furthermore, as $||P|| = 1$ for all ϕ, $||E|| = 1$ so that E is a quaternion volume by definition 8.1 with $n = 2$.

Similarly, when both P and Q are quaternion volumes the elements of E are sums of products of $p_i(\phi_1)q_j(\phi_2)$ and E is a quaternion volume by the same argumentation. The same argumentation applies when P and Q are quaternions or quaternion volumes with more than one restriction, $n > 1$. □

8.4.2 Quaternion Volumes by Rotations Sequences

A rotation sequence describes a rotation about one coordinate axis followed by a rotation about another of the coordinate axes in the rotated coordinate system. A general framework on how to construct easily visualisable quaternion volumes by rotation sequences is presented. The rotation sequence starts with two subsequent rotations about two coordinate axes, represented by the quaternion Q_s. This defines the attitude direction. The last degree of freedom is added by a rotation about the direction vector, here the z-axis, by Q_z. In equation (8.6), let Q_z represent the vector to be rotated and let Q_s be the quaternion describing the direction of this vector. Then the rotation sequence

$$\mathcal{V} = Q_s * Q_z * Q_s^* \tag{8.15}$$

represents the direction of the z-axis for a given rotation Q_s given by the direction of the vector part of $\mathcal{V}$ and the rotation about the z-axis given by the scalar part or length of the vector part of $\mathcal{V}$ by $\psi = 2\arcsin(||\overline{\boldsymbol{v}}||) = 2\arccos(v_0)\text{sgn}(\psi)$. Henceforth, $\mathcal{V}$ is called a *visualising quaternion*. Note that $\mathcal{V}$ does not represent a rotation. It is used as a tool to visualise rotations and as a help to define an appropriate set of frames for different applications. The visualising quaternion and the corresponding quaternion should be viewed upon as a pair, $(Q, \mathcal{V})$, where the visualising quaternion, $\mathcal{V}$, gives an intuitive description of a rotation of a frame by Q.

Let the vector part of the visualising quaternion be plotted as a point in the xyz-sphere. Then the direction of the z-axis, rotated by the corresponding quaternion is given by the vector from the origin to this point, and the rotation about the z-axis itself is given by the length of this vector. Hence, a continuous set of quaternions (a quaternion volume) is represented by a "cloud" in the xyz-sphere describing the corresponding set of orientations.

The quaternion that rotates the reference frame into the orientation described by equation (8.15) is then given by

$$Q = Q_z * Q_s. \tag{8.16}$$

Finally, the quaternion volume is given by restricting the allowed rotations of each quaternion.

[2]The real numbers are closed under addition and multiplication, hence if a and b are real numbers, so are a+b and ab (Kuipers, 2002).

Given a visualising quaternion volume by the sequence

$$\mathcal{V}^{\otimes} = Q_s^{\otimes} * Q_z^{\otimes} * (Q_s^{\otimes})^* \tag{8.17}$$

and the restrictions on $Q_s^{\otimes}$ and $Q_z^{\otimes}$. Then the corresponding quaternion volume that results in the set of orientations described by $\mathcal{V}^{\otimes}$ is given by

$$Q^{\otimes} = Q_z^{\otimes} * Q_s^{\otimes} \tag{8.18}$$

with the same restrictions applied to $Q^{\otimes}$ as to $\mathcal{V}^{\otimes}$.

Figure 8.1 shows the difference between the quaternion volume and the visualising quaternion volume when the vector part is plotted in the xyz-sphere[3]. Note that the dual covering also applies to the visualising quaternion volume. Hence, one should always keep track of the sign of the rotation so that a negative rotation about the z-axis is not interpreted as an opposite direction of the z-axis. This can be done by moving the negative sign to the scalar part (which is positive in $(-\pi, \pi)$ or to assume all angles in the interval $(0, 2\pi)$ where $\sin(\frac{\phi}{2})$ is positive.

8.4.3 Reorientation of Quaternion Volumes

Let $Q^{\otimes}$ be a quaternion volume and the quaternion P represent some transformation on $Q^{\otimes}$. It will be claimed that the transformation $Q_P^{\otimes} = Q^{\otimes} * P$ rotates the entire set of frames by a rotation P. Similarly, the transformation $Q_P^{\otimes} = Q^{\otimes} * P^*$ allows the set of frames represented by the quaternion volume to be represented with respect to a new reference frame P. The transformation induced by changing from one reference orientation to another is called *reorientation* (Alpern et al., 1993).

Proposition 8.2 (Transformation of Quaternion Volumes)**.** *Any quaternion volume, $Q^{\otimes}$, represented with respect to the identity frame can be transformed into another quaternion volume by*

$$Q_P^{\otimes} = Q^{\otimes} * P, \tag{8.19}$$

where the orientations represented by $Q_P^{\otimes}$ relate to P in the same way as $Q^{\otimes}$ relates to the identity frame.

Proof. The quaternion product $E = Q * P$ can be viewed upon as a rotation P followed by a rotation Q with respect to the *new* frame that resulted from the first rotation P. Hence, E relates to P in the same way as Q relates to the identity frame. By the same argumentation the quaternion volume $Q_P^{\otimes}$ relates to P in the same way as $Q^{\otimes}$ relates to the identity frame. □

In proposition 8.2, the reference frame is kept constant and all the elements of the quaternion volume are rotated by P. Reorientation, however, is a rotation of the reference frame (change of observer) while the quaternion volume is kept constant. The proof of the reorientation $Q_P^{\otimes} = Q^{\otimes} * P^*$ is constructed in the same way as the proof of proposition 8.2.

[3]In figure 8.1, the orientations are plotted by $\sin(\frac{\phi}{2})\boldsymbol{n}$ (the orthographic orientation map (Alpern et al., 1993)) for both the quaternion volume and the visualising quaternion volume.

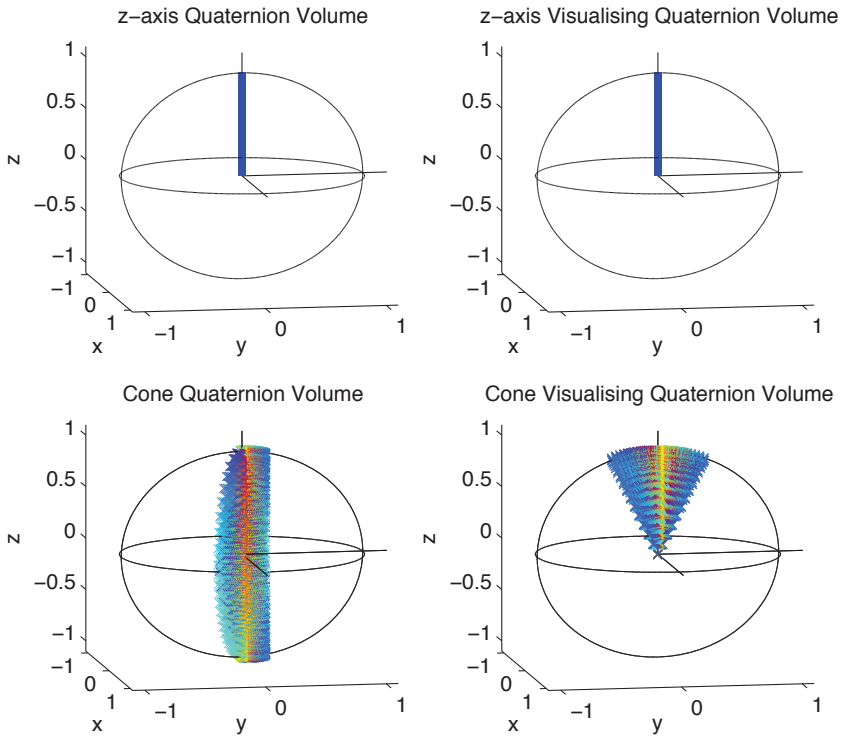


Figure 8.1: The quaternion volume and the visualising quaternion volume in the xyz-sphere. The upper plots show a freedom about the z-axis and the lower plots show all vectors that span out a cone and the orientations about these vectors. The visualising quaternion volume gives a more intuitive picture of the orientations described by the quaternion volume than the quaternion volume itself when plotted in the xyz-sphere.

Comment 1. From equations (8.3) and (8.6), four different ways of transforming quaternion volumes arise.

$$\begin{aligned} &1)\ Q^{\otimes}_{P_1} = Q^{\otimes} * P \qquad && 3)\ Q^{\otimes}_{P_3} = Q^{\otimes} * P * (Q^{\otimes})^* \\ &2)\ Q^{\otimes}_{P_2} = P * Q^{\otimes} && 4)\ Q^{\otimes}_{P_4} = P * Q^{\otimes} * P^* \end{aligned}$$

The first transformation is used in Hanson (2006) to find a set of frames, all with one axis pointing in a fixed direction, as a mean to find an optimal path in the quaternion space. If $Q^{\otimes}$ represents a freedom about one of the coordinate axes, say the z-axis, $Q^{\otimes}_{P_1}$ can also represent a set of orientations where the z-axes have the same angles with respect to the reference frame z-axis, determined by P. For this special case, the same result is obtained by the third representation. Even though the two representations present the same set of vectors they differ in orientation. The set of frames described by $Q^{\otimes}_{P_1}$ is the set that results from rotating the frame represented by P about the coordinate axis, while $Q^{\otimes}_{P_3}$ is the set

of frames when the shortest rotation is taken from the reference frame to the directions described by P and $Q^{\otimes}$.

8.5 Coordinate Axis Rotation

There are several ways of representing the proximity of two frames (Yuan, 1988; Wen and Kreutz-Delgado, 1991). Here, the proximity of two frames will be described by the rotation required to take one frame into the other by the shortest rotation.

Definition 8.2 (Quaternion Space Proximity)**.** Given two orientations represented by the two quaternions P and Q. Let the *error quaternion* be denoted

$$E = P^* * Q. \tag{8.20}$$

Then the scalar part of E, e_0, describes the proximity of the two frames.

Definition 8.3 (Minimal Rotation)**.** The larger (closer to 1[4]) the error quaternion scalar part e_0, the closer are the two orientations P and Q.

An uncountable number of devices have only one degree of rotational freedom, ranging from human elbows and revolute robotic joints to satellites with only one operating actuator. The control of a one-actuator satellite is important whenever actuator failure occurs. Two questions arise:

1. How close to the desired orientation can one get with just one degree of freedom.
2. What is the reachable orientation closest to the desired orientation.

Assume that $P_z^{\otimes}$ represents the set of orientations when the identity frame is rotated about the z-axis. Then the problem is to find the P_z that that is closest to Q.

Proposition 8.3 (Optimal Rotation)**.** *Consider an orientation* $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^{\mathsf{T}}$. *The orientation described by the quaternion* $P_z = \begin{bmatrix} p_0 & 0 & 0 & p_3 \end{bmatrix}^{\mathsf{T}}$ *that is closest to* Q *(by definition 8.2 and 8.3) is given by*

$$p_0 = \frac{\pm_s q_0}{\sqrt{q_0^2 + q_3^2}} \tag{8.21}$$

$$p_3 = \frac{\pm_s q_3}{\sqrt{q_0^2 + q_3^2}} \tag{8.22}$$

where the two $\pm_s$ *have the same sign.*

Proof. $E = P^* * Q$ can be written

$$\begin{bmatrix} e_0 \\ e_3 \end{bmatrix} = \begin{bmatrix} p_0 & p_3 \\ -p_3 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_3 \end{bmatrix} \tag{8.23}$$

[4]Note that an equally good description of proximity is given when e_0 approaches -1. As $\cos(\frac{\phi}{2})$ is positive for ϕ in the chosen interval $(-\pi, \pi)$, the positive value of e_0 is chosen.

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} p_0 & p_3 \\ -p_3 & p_0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \tag{8.24}$$

By definitions 8.2 and 8.3, the quaternion P_z that is closest to Q is found by the error quaternion with e_0 closest to 1.

$$e_0 = p_0 q_0 + p_3 q_3 \tag{8.25}$$

$$= q_0 \cos(\frac{\psi}{2}) + q_3 \sin(\frac{\psi}{2}), \tag{8.26}$$

so that

$$\frac{\mathrm{d}e_0}{\mathrm{d}\psi} = -\frac{q_0}{2} \sin(\frac{\psi}{2}) + \frac{q_3}{2} \cos(\frac{\psi}{2}). \tag{8.27}$$

Let $\frac{\mathrm{d}e_0}{\mathrm{d}\psi} = 0$. Then

$$\tan(\frac{\psi}{2}) = \frac{q_3}{q_0}. \tag{8.28}$$

Then by using $\arctan(x) = \arcsin\left(\frac{x}{\sqrt{1+x^2}}\right)$ (Bronshtein et al., 2003), ψ is written as

$$\psi = 2 \arctan(\frac{q_3}{q_0}) \tag{8.29}$$

$$= 2 \arcsin\left(\frac{\frac{q_3}{q_0}}{\sqrt{1 + (\frac{q_3}{q_0})^2}}\right) \tag{8.30}$$

$$= 2 \arcsin\left(\frac{q_3}{\sqrt{q_0^2 + q_3^2}}\right). \tag{8.31}$$

From the definition of the quaternion

$$\psi = 2 \arcsin(p_3). \tag{8.32}$$

By comparing equations (8.31) and (8.32), equation (8.22) is given. Similarly by $\arctan(x) = \arccos\left(\frac{1}{\sqrt{1+x^2}}\right) \mathrm{sgn}(x)$

$$\psi = 2 \arctan(\frac{q_3}{q_0}) \tag{8.33}$$

$$= 2 \arccos\left(\frac{1}{\sqrt{1 + (\frac{q_3}{q_0})^2}}\right) \mathrm{sgn}(\frac{q_3}{q_0}) \tag{8.34}$$

$$= 2 \arccos\left(\frac{q_0}{\sqrt{q_0^2 + q_3^2}}\right) \mathrm{sgn}(\frac{q_3}{q_0}). \tag{8.35}$$

Note that the sign of $\psi = 2 \arccos(p_0)\mathrm{sgn}(\psi)$ is given by equation (8.32). Hence, equation (8.21) is found. For ψ to be in the interval $[-\pi, \pi]$, the sign $\pm_s$ is chosen positive, so that e_0 is positive. Similarly when P rotates about the x- and y-axis. □

The largest rotation is given when e_0 is close to zero.

$$e_0 = p_0 q_0 + p_3 q_3 \tag{8.36}$$

$$= q_0 \cos(\frac{\psi}{2}) + q_3 \sin(\frac{\psi}{2}) = 0. \tag{8.37}$$

$$\tan(\frac{\psi}{2}) = -\frac{q_0}{q_3}. \tag{8.38}$$

Similar to the proof of proposition 8.3, the orientation P_z furthest away from Q is given by

$$p_0 = \frac{\pm_s q_3}{\sqrt{q_0^2 + q_3^2}} \tag{8.39}$$

$$p_3 = \frac{\pm_t q_0}{\sqrt{q_0^2 + q_3^2}} \tag{8.40}$$

where the $\pm_s$ and $\pm_t$ have opposite signs.

8.6 Quaternion Volume Desired Attitude

In the following, a satellite is used to illustrate the results presented above and how they apply to the control of rigid bodies. Two basic problems are addressed in this paper.

1. Fuel consumption is critical in the control of satellites. A methodology on how to represent the desired attitude by a quaternion volume as a mean to save energy is proposed.

2. Failure in one or more of the satellite actuators greatly complicates the control and can result in a desired attitude that is out of reach. A method on how to take the satellite as close as possible to the desired attitude with just one actuator is proposed.

A satellite with three actuators is considered. Each actuator applies a torque about one of the coordinate axes of the satellite body frame.

The body frame and desired attitude are defined with respect to the North-East-Down coordinate system (NED-frame). The x-axis of the body frame points in the speed direction and the operating device (camera, telescope, transponder, etc) is aligned along the body frame z-axis.

8.6.1 Desired Attitude Direction

First assume that the satellite attitude must be aligned such that the z-axis is always orthogonal to the earth's surface, pointing towards the earth. This gives the satellite one degree of freedom about the z-axis. An arbitrary rotation, ψ, about the z-axis can be represented by the quaternion volume

$$Q_{free}^{\otimes} = \begin{bmatrix} \cos(\frac{\psi}{2}) & 0 & 0 & \sin(\frac{\psi}{2}) \end{bmatrix}^{\mathsf{T}}, \quad \text{for} \quad -\pi < \psi < \pi. \tag{8.41}$$

Hence, the quaternion volume is given with respect to the identity frame. Further, assume the desired quaternion volume instead is to be rotated by $Q_d = \begin{bmatrix} d_0 & d_1 & d_2 & d_3 \end{bmatrix}^{\mathsf{T}}$ with

respect to the identity frame. The quaternion volume that describes all attitudes where the z-axis points in the same direction as the z-axis of Q_d is given by $Q_d^{\otimes} = Q_{free}^{\otimes} * Q_d$ so that

$$Q_d^{\otimes} = \begin{bmatrix} d_0\cos(\frac{\psi}{2}) - d_3\sin(\frac{\psi}{2}) \\ d_1\cos(\frac{\psi}{2}) - d_2\sin(\frac{\psi}{2}) \\ d_2\cos(\frac{\psi}{2}) + d_1\sin(\frac{\psi}{2}) \\ d_3\cos(\frac{\psi}{2}) + d_0\sin(\frac{\psi}{2}) \end{bmatrix}, \quad \text{for} \quad -\pi < \psi < \pi. \tag{8.42}$$

Example 7. If the desired orientation is set so that the z-axis is always orthogonal to the xy-plane, pointing outwards into space, by a rotation about the y-axis, $Q_d = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^{\mathsf{T}}$, equation (8.42) simplifies to

$$Q_d^{\otimes} = \begin{bmatrix} 0 & -\sin(\frac{\psi}{2}) & \cos(\frac{\psi}{2}) & 0 \end{bmatrix}^{\mathsf{T}}, \quad \text{for} \quad -\pi < \psi < \pi. \tag{8.43}$$

It can be shown that this quaternion volume represents all attitudes with a z-axis in the opposite direction of the NED-frame z-axis.

$$\begin{aligned} \boldsymbol{z}^{\otimes} &= Q_d^{\otimes} * \boldsymbol{v}_z * (Q_d^{\otimes})^* \qquad (8.44) \\ &= \begin{bmatrix} 0 \\ -\sin(\frac{\psi}{2}) \\ \cos(\frac{\psi}{2}) \\ 0 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} * \begin{bmatrix} 0 \\ \sin(\frac{\psi}{2}) \\ -\cos(\frac{\psi}{2}) \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} -\cos(\frac{\psi}{2})\sin(\frac{\psi}{2}) + \cos(\frac{\psi}{2})\sin(\frac{\psi}{2}) \\ 0 \\ 0 \\ -\cos^2(\frac{\psi}{2}) - \sin^2(\frac{\psi}{2}) \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}, \quad \text{for} \quad -\pi < \psi < \pi. \qquad (8.45) \end{aligned}$$

8.6.2 Desired Attitude Orientation

The attitude can be represented as a set of frames. This set can be composed by a rotation sequence of quaternion volumes. Two rotation sequences are discussed, the ZYZYZ-sequence, as in Hanson (2006) and Alpern et al. (1993), and the XYZYX-sequence.

The ZYZYZ-sequence

The ZYZYZ-sequence allows the desired attitude to be defined as a set of vectors that span out a cone about the reference z-axis and all orientations about these vectors. Let $Q_s(\alpha, \beta) = Q(\beta, \boldsymbol{y}) * Q(\alpha, \boldsymbol{z})$ where $Q(\alpha, \boldsymbol{z}) = \begin{bmatrix} \cos(\frac{\alpha}{2}) & 0 & 0 & \sin(\frac{\alpha}{2}) \end{bmatrix}^{\mathsf{T}}$ and $Q(\beta, \boldsymbol{y}) =$

$\begin{bmatrix}\cos(\frac{\beta}{2}) & 0 & \sin(\frac{\beta}{2}) & 0\end{bmatrix}^{\mathsf{T}}$ so that

$$Q_s(\alpha,\beta) = \begin{bmatrix}\cos(\frac{\alpha}{2})\cos(\frac{\beta}{2}) \\ \sin(\frac{\alpha}{2})\sin(\frac{\beta}{2}) \\ \cos(\frac{\alpha}{2})\sin(\frac{\beta}{2}) \\ \sin(\frac{\alpha}{2})\cos(\frac{\beta}{2})\end{bmatrix}. \tag{8.46}$$

The quaternion volume can be visualised in the xyz-sphere (see figure 8.1) by the three last elements of

$$\mathcal{V}^{\otimes}(\alpha,\beta,\gamma) = \begin{bmatrix}\cos(\frac{\gamma}{2}) \\ \sin(\frac{\gamma}{2})\sin(\beta)\cos(\alpha) \\ \sin(\frac{\gamma}{2})\sin(\alpha)\sin(\beta) \\ \sin(\frac{\gamma}{2})\cos(\beta)\end{bmatrix}. \tag{8.47}$$

α represents the allowed orientations about the z-axis of the first rotation while β is the allowed orientation about the new y-axis. If α has no restrictions, β is the offset angle that defines a cone with the z-axis at the centre. γ restricts the orientation about the z-axis itself. A cone sector that allows a deviation, b_{max}, in the sector defined by the restrictions on α in the xy-plane is defined by

$$Q_d^{\otimes} = Q_z^{\otimes} * Q_s^{\otimes} = \begin{bmatrix}\cos(\frac{\beta}{2})\cos(\frac{\alpha}{2}+\frac{\gamma}{2}) \\ \sin(\frac{\beta}{2})\sin(\frac{\alpha}{2}-\frac{\gamma}{2}) \\ \sin(\frac{\beta}{2})\cos(\frac{\alpha}{2}-\frac{\gamma}{2}) \\ \cos(\frac{\beta}{2})\sin(\frac{\alpha}{2}+\frac{\gamma}{2})\end{bmatrix} \tag{8.48}$$

and the restrictions

$$a_{min} \leq \alpha \leq a_{max} \tag{8.49}$$
$$0 \leq \beta \leq b_{max} \tag{8.50}$$
$$c_{min} \leq \gamma \leq c_{max} \tag{8.51}$$

Example 8. Assume a satellite where the z-axis is to point outwards into space. Further assume that a small error, b_{max}, in the orientation is allowed and only the attitude directions are restricted. The set of frames describing these attitudes is given by (8.48) and the restrictions

$$-\pi \leq \alpha \leq \pi \tag{8.52}$$
$$\pi \leq \beta \leq \pi + b_{max} \tag{8.53}$$
$$-\pi \leq \gamma \leq \pi \tag{8.54}$$

It can be showed that this is the same as substituting
$\beta \leftarrow \pi + \beta$ and $\alpha \leftarrow -\alpha$ into equation (8.48) so that

$$Q_d^{\otimes} = Q_s^{\otimes} * Q_z^{\otimes} = \begin{bmatrix}-\sin(\frac{\beta}{2})\cos(\frac{\alpha}{2}-\frac{\gamma}{2}) \\ \cos(\frac{\beta}{2})\sin(\frac{\alpha}{2}+\frac{\gamma}{2}) \\ \cos(\frac{\beta}{2})\cos(\frac{\alpha}{2}+\frac{\gamma}{2}) \\ -\sin(\frac{\beta}{2})\sin(\frac{\alpha}{2}-\frac{\gamma}{2})\end{bmatrix} \tag{8.55}$$

and keeping the restrictions (8.49)-(8.51). Note that equation (8.55) can also be obtained by rotating the quaternion volume of the previous example by π radians about the y-axis, hence by equation (8.19) with $P = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^\mathsf{T}$ and $Q^\otimes = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ as in (8.48) so that $Q_d^\otimes = \begin{bmatrix} -q_2 & q_3 & q_0 & -q_1 \end{bmatrix}^\mathsf{T}$, which is the same as equation (8.55).

The XYZYX-sequence

The XYZYX-sequence defines a pyramid of allowed orientations where the allowed orientations about the x-axis and the (new) y-axis are restricted. This is a good estimation of restricting the orientation about the globally defined x- and y-axes whenever the angles are kept small. $Q_s(\alpha, \beta)$ is then given by

$$Q_s(\alpha, \beta) = \begin{bmatrix} \cos(\frac{\alpha}{2})\cos(\frac{\beta}{2}) \\ \sin(\frac{\alpha}{2})\cos(\frac{\beta}{2}) \\ \cos(\frac{\alpha}{2})\sin(\frac{\beta}{2}) \\ -\sin(\frac{\alpha}{2})\sin(\frac{\beta}{2}) \end{bmatrix}, \tag{8.56}$$

and visualised by

$$\mathcal{V}^\otimes(\alpha, \beta, \gamma) = \begin{bmatrix} \cos(\frac{\gamma}{2}) \\ \sin(\frac{\gamma}{2})\sin(\beta) \\ -\sin(\frac{\gamma}{2})\sin(\alpha)\cos(\beta) \\ \sin(\frac{\gamma}{2})\cos(\alpha)\cos(\beta) \end{bmatrix}. \tag{8.57}$$

The corresponding quaternion volume is again given by

$$Q_d^\otimes = Q_z^\otimes * Q_s^\otimes \tag{8.58}$$

and the restrictions

$$a_{min} \leq \alpha \leq a_{max} \tag{8.59}$$
$$b_{min} \leq \beta \leq b_{max} \tag{8.60}$$
$$c_{min} \leq \gamma \leq c_{max} \tag{8.61}$$

Example 9. Assume a satellite where the attitude is to be restricted similarly to example 8, but instead of allowing some error in the orientation, an orientation error about the x- and y-axes are restricted to $\pm a$ and $\pm b$. Then the set of frames describing these attitudes is given by (8.58) and the restrictions

$$\pi - a \leq \alpha \leq \pi + a \tag{8.62}$$
$$-b \leq \beta \leq b \tag{8.63}$$
$$-\pi \leq \gamma \leq \pi \tag{8.64}$$

8.7 Control

Two ways to exploit the quaternion volume representation to reduce fuel consumption are presented.

1. Let the desired attitude (one frame only) take part in a control loop. When the attitude is inside the attitude specifications given by the quaternion volume, some action is taken to save energy. This may be to switch to another controller which requires less energy or to switch to another desired attitude inside the quaternion volume, closer or equal to the current attitude. Note also that if the quaternion volume defines a set of orientations close to some reference orientation, a linearised model of the satellite may be used.
2. Find the frame within the set of frames restricted by the quaternion volume that corresponds to the shortest rotation from the current orientation and set this as the desired attitude.

Two problems arise.

1. A test to verify if a frame is inside the quaternion volume is needed.
2. Find the orientation inside the quaternion volume that results in the shortest rotation from the current orientation.

8.7.1 Quaternion Volume Test

Consider a quaternion volume defined by the ZYZYZ-sequence. A test to verify if a query quaternion $Q_{qry} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ is an element of the quaternion volume is presented.

A query quaternion can be represented in terms of α, β and γ. The transformation between the quaternion representation and the (α,β,γ)-representation can be performed in many ways, by geometric analysis, by the visualising quaternion or through a quaternion/orientation map. The first method is often the easiest and most intuitive method and works well when only the direction is concerned. When the full orientation is to be determined, this approach is not suitable. In the following, it is shown how this method can be combined with the visualising quaternion to find the orientation.

By noting that α and β can be seen from the direction of the z-axis only (not from the entire frame), they can be found from the vector $\bar{\boldsymbol{z}} = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T}$ of the rotation of the vector along the z-axis, $\bar{\boldsymbol{v}}_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\mathsf{T}$ by $\boldsymbol{z} = Q_{qry} * \boldsymbol{v}_z * Q^*_{qry}$. Then, by standard geometrical relations α and β are found.

$$\alpha = \text{arctan2}\left(\frac{y}{x}\right), \tag{8.65}$$

$$\beta = \arccos(z), \tag{8.66}$$

where x, y and z are the elements of $\bar{\boldsymbol{z}}$ given by

$$\bar{\boldsymbol{z}} = \begin{bmatrix} 2q_0q_2 + 2q_1q_3 \\ 2q_2q_3 - 2q_0q_1 \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{8.67}$$

As already stated, the rotation about the z-axis cannot be seen from the vector rotation of the z-axis, but is found from $\mathcal{V}$ by $\gamma = 2\arccos(v_0)$. The sign of γ is lost in the transformation but can be found by, for example the sign of the fourth element of $\mathcal{V}$.

Given a query quaternion Q_{qry}. Then α, β and γ from the ZYZYZ-sequence are found by

$$\alpha = \arctan2\left(\frac{q_2q_3 - q_0q_1}{q_0q_2 + q_1q_3}\right), \tag{8.68}$$

$$\beta = \arccos(q_0^2 - q_1^2 - q_2^2 + q_3^2), \tag{8.69}$$

$$\gamma = 2\arccos(v_0)\,\mathrm{sgn}(v_4). \tag{8.70}$$

It is now straight forward to verify if the quaternion is inside the quaternion volume.

The quaternion volume is not always of such a structure that it can be analysed geometrically. Then the analytic expression of the quaternion volume can be used. For comparison, this approach is also shown for the ZYZYZ-sequence.

The quaternion volume is given by equation (8.48) and its restrictions.

$$\begin{bmatrix} \cos(\frac{\beta}{2})\cos(\frac{\alpha}{2}+\frac{\gamma}{2}) \\ \sin(\frac{\beta}{2})\sin(\frac{\alpha}{2}-\frac{\gamma}{2}) \\ \sin(\frac{\beta}{2})\cos(\frac{\alpha}{2}-\frac{\gamma}{2}) \\ \cos(\frac{\beta}{2})\sin(\frac{\alpha}{2}+\frac{\gamma}{2}) \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \begin{matrix} (I) \\ (II) \\ (III) \\ (IV) \end{matrix} \tag{8.71}$$

By substituting (II) into (III), (III) simplifies to

$$\sin(\frac{\beta}{2})\sqrt{1 - \frac{q_1^2}{\sin^2(\frac{\beta}{2})}} = q_2 \tag{8.72}$$

so that β is found by

$$\beta = 2\arcsin\sqrt{q_1^2 + q_2^2}. \tag{8.73}$$

β is positive by definition. α and γ are found by dividing (II) by (III) and (IV) by (I):

$$\tan(\frac{\gamma}{2} - \frac{\alpha}{2}) = \frac{q_1}{q_2}, \tag{8.74}$$

$$\tan(\frac{\gamma}{2} + \frac{\alpha}{2}) = \frac{q_3}{q_0}. \tag{8.75}$$

Further let

$$\frac{\gamma}{2} - \frac{\alpha}{2} = \arctan(\frac{q_1}{q_2}), \tag{8.76}$$

$$\frac{\gamma}{2} + \frac{\alpha}{2} = \arctan(\frac{q_3}{q_0}). \tag{8.77}$$

so that α and γ are given by

$$\alpha = \arctan(\frac{q_3}{q_0}) - \arctan(\frac{q_1}{q_2}), \tag{8.78}$$

$$\gamma = \arctan(\frac{q_3}{q_0}) + \arctan(\frac{q_1}{q_2}). \tag{8.79}$$

Hence, α and β can by found by geometrical interpretation while α, β and γ are found from the analytical expression of the quaternion volume. In the following, it is shown that these two approaches give the same result. From equation (8.69), β is simplified by

$$\beta = \arccos(q_0^2 - q_1^2 - q_2^2 + q_3^2) \tag{8.80}$$

$$= \arccos(2(q_0^2 + q_3^2) - 1) \tag{8.81}$$

and the trigonometric relation $2\arccos(x) = \arccos(2x^2 - 1)$ (Bronshtein et al., 2003) so that

$$\beta = 2\arccos\sqrt{q_0^2 + q_3^2}. \tag{8.82}$$

By $\arccos(x) = \arcsin(\sqrt{1-x^2})$ this is equal to equation (8.73). By $\arctan(x) + \arctan(y) = \arctan2\left(\frac{x+y}{1-xy}\right)$, equation (8.78) can be written as

$$\alpha = \arctan(\frac{q_3}{q_0}) - \arctan(\frac{q_1}{q_2}) \tag{8.83}$$

$$= \arctan2\left(\frac{\frac{q_3}{q_0} - \frac{q_1}{q_2}}{1 + \frac{q_1 q_3}{q_0 q_2}}\right) \tag{8.84}$$

$$= \arctan2\left(\frac{q_2 q_3 - q_0 q_1}{q_0 q_2 + q_1 q_3}\right). \tag{8.85}$$

γ can be written in the same way so that, alternatively, a complete description of the query quaternion can be given by

Given a query quaternion Q_{qry}. Then α, β and γ from the ZYZYZ-sequence are found by

$$\alpha = \arctan2\left(\frac{q_2 q_3 - q_0 q_1}{q_0 q_2 + q_1 q_3}\right), \tag{8.86}$$

$$\beta = 2\arccos\sqrt{q_0^2 + q_3^2}, \tag{8.87}$$

$$\gamma = \arctan2\left(\frac{q_2 q_3 + q_0 q_1}{q_0 q_2 - q_1 q_3}\right). \tag{8.88}$$

8.7.2 Transformed Quaternion Volumes

The easiest way to verify if a query quaternion is inside a quaternion volume transformed by equation (8.19) is to transform the query quaternion by the opposite transformation, P, so that both the quaternion volume and the query quaternion are presented in the identity frame. Hence, the two problems below are identical.

$$Q_{qry} \in P * Q^{\otimes} \quad ? \tag{8.89}$$

$$P^* * Q_{qry} \in Q^{\otimes} \quad ? \tag{8.90}$$

This operation is computationally demanding. As equation (8.19) gives an analytical expression of the transformed quaternion volume, the orientation should be found by a set of

parameters similar to the ones found in equations (8.86)-(8.88). This may be done when the quaternion volume is on a simple form, for example by equation (8.55). Then the query quaternion may be tested against the restrictions in (8.49)-(8.51) directly. By following the mathematics of equations (8.71)-(8.88), α, β and γ are found with respect to the coordinate system of $P = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^\mathsf{T}$ by

$$\alpha_P = \arctan2\left(\frac{q_0 q_1 - q_2 q_3}{q_0 q_2 + q_1 q_3}\right), \tag{8.91}$$

$$\beta_P = 2\arcsin\sqrt{q_0^2 + q_3^2}, \tag{8.92}$$

$$\gamma_P = \arctan2\left(\frac{q_0 q_1 + q_2 q_3}{q_0 q_2 - q_1 q_3}\right). \tag{8.93}$$

Hence, as expected

$$\beta_P = \beta - \pi, \tag{8.94}$$

$$\alpha_P = -\alpha, \tag{8.95}$$

$$\gamma_P = \gamma. \tag{8.96}$$

8.7.3 Clamping

If any of the restrictions are violated, the quaternion might be clamped into the set of frames restricted by the quaternion volume in many ways. Hanson (2006) suggests finding the nearest point in the quaternion metric. Another intuitively tempting approach is to set the exceeded value to the maximum allowed value. Then a quaternion that is inside the quaternion volume may be constructed by the definition in equation (8.48) directly. If the quaternion volume is on a simple form, the orientation can be clamped into the quaternion volume by the shortest rotation in order to save energy.

8.7.4 Shortest Rotation

How to find the orientation in the quaternion volume that results in the shortest rotation from the current attitude depends on the quaternion volume. One simple solution occurs when the quaternion volume represents a freedom about one axis. Then the theory from section 8.5 can be applied directly. Let Q be the current attitude and $P_{free}^{\otimes} = \begin{bmatrix} \cos\left(\frac{\psi}{2}\right) & 0 & 0 & \sin\left(\frac{\psi}{2}\right) \end{bmatrix}^\mathsf{T}$ represent the set of allowed attitudes, both defined in the NED-frame. Then the attitude within the quaternion volume that is closest to the current attitude is given by

$$p_0 = \frac{\pm_s q_0}{\sqrt{q_0^2 + q_3^2}} \tag{8.97}$$

$$p_3 = \frac{\pm_s q_3}{\sqrt{q_0^2 + q_3^2}} \tag{8.98}$$

and the rotation required to take Q into P (the error) is given by

$$\psi = 2\arccos(e_0), \tag{8.99}$$

where $e_0 = p_0 q_0 + p_3 q_3$.

8.7.5 Closest Orientation

Closely related to the problem of the previous subsection is the problem of finding the optimal rotation when actuator failure occurs. When only one actuator is functional, the satellite can only rotate about one of the axes of the body frame. Let the current body orientation be described in the NED-frame by T. The desired orientation, Q_d, is also given in the NED-frame. Assume that the freedom, represented by a quaternion volume, $Q^{\otimes}_{free}$, is given with respect to the body frame, so that

$$Q^{\otimes}_{T} = T * Q^{\otimes}_{free}. \tag{8.100}$$

represent all reachable orientations. Then the problem is to find the quaternion $Q \in Q^{\otimes}_{free}$ that takes the satellite as close to the desired orientation as possible. This is given by proposition 8.3 where E is given by

$$E = Q^{*}_{d} * Q^{\otimes}_{T} \tag{8.101}$$

$$= Q^{*}_{d} * T * Q^{\otimes}_{free}. \tag{8.102}$$

Let $T^{*}_{d} = Q^{*}_{d} * T = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{bmatrix}$. Then, as only the size (not the direction) of the rotation is considered, the closest possible orientation is given by equations (8.21) and (8.22) and the rotation needed to take the satellite from the closest reachable orientation to the desired orientation (the error) is given by equation (8.99).

8.8 Conclusions

The unit quaternion group is used to find a general framework for representing sets of orientations. It is also shown how this set can be represented with respect to another reference frame or how to rotate this set when the reference frame is kept constant. Several examples of sets of orientations are presented and it is shown how these sets can be represented by a quaternion and some easy to visualise restrictions. A satellite is used to illustrate how to save energy by defining the desired attitude as a set of orientations. A method to verify whether a quaternion is inside a quaternion volume is also presented. It is also shown how to find the rotation that requires less energy in order to take the desired attitude into an element of the quaternion volume. Some work is done on finding the orientation closest to the desired orientation when the desired orientation is out of reach.

Chapter 9

On the Equivalence of Orientation Error and Positive Definiteness of Matrices

9.1 Abstract

In this paper we show how a continuous set of orientations can be represented as a positive definiteness test on a given matrix. When this continuous set is restricted by the maximum allowed orientation error in some or all directions it is shown that the requirement for an orientation to satisfy these restrictions is equivalent to positive definiteness for a certain matrix. The problem of finding the optimal orientation that satisfies these restrictions is hence transformed into an optimisation problem on the Riemannian manifold of linearly constrained symmetric positive definite matrices. Thus, the problem of finding the optimal orientation can be solved as a standard optimisation problem with the constraints written in the form of linear matrix inequalities or barrier functions. Linear matrix inequalities have been extensively studied in the optimisation communities and good and efficient algorithms are available.

9.2 Introduction

In a wide range of applications the orientation of a rigid body does not need to be restricted to one frame but can be given as a continuous set of frames. The attitude of a satellite can for example be set so that the transmitter or receiver points approximately in the direction of the earth. Another example is the end effector of a robotic manipulator where an orientation error is allowed in the end-effector orientation. In Potkonjak et al. (2000) the idea of introducing the paint quality as a constraint and minimise some additional cost function was presented. This opens for the possibility of allowing an orientation error in the orientation of the end effector in order to improve the speed of the job, reduce torques and so on. It was shown in From and Gravdahl (2007a) that by allowing an orientation error in

the end effector configuration of a robotic manipulator, the speed and the quality of the job was improved. The orientation error was chosen intuitively, and the approach presented was not suitable for implementation in an optimisation algorithm. The pointing task is another example where a continuous set of orientations is allowed in the specifications of the end-effector orientation.

In Buss et al. (1996) the problem of friction force limit constraints is transformed into a problem of testing for positive definiteness of a certain matrix. In this paper the same ideas are used to convert the problem of orientation error constraints into a test of positive definiteness of a matrix. For different types of orientation errors, a suitable matrix is found and it is shown that positive definiteness of this matrix is equivalent to an orientation that satisfies the given restrictions on the orientation.

By transforming the nonlinear orientation constraints into positive definiteness constraints imposed on certain symmetric matrices the problem of finding the optimal orientation is transformed into an optimisation problem on the smooth manifold of linearly constrained positive definite matrices. For the special case of positive definite symmetric matrices, the problem can be transformed into solving a linear matrix inequality (LMI). Convex optimisation problems involving LMIs have been extensively studied in literature, and good solutions, such as interior point algorithms, are known.

For many sets of orientations a symmetric matrix can be found very easily. For other sets with a more complicated structure, a symmetric matrix may be hard to find. It is a clear advantage to choose the sets for which a symmetric matrix can be found because LMIs of symmetric matrices are in general solved very efficiently. The difference between the sets for which a symmetric matrix can and cannot be found is shown through the examples.

The applications range from satellites and aircraft to robotics and rigid bodies in general. A satellite acted upon by an external force can be written as an LMI with an additional linear constraint. The cost function will typically tend to infinity at the border of positive definiteness, which is equivalent to orientation error constraints. In robotics, the freedom represented by the orientation can be used to improve the performance as in From and Gravdahl (2007a). The advantage of the approach presented here is that the problem can be formulated as an optimisation problem on a smooth manifold. Hence, an optimal solution can be found, as opposed to the intuitively found geometric solution presented in From and Gravdahl (2007a).

The problems considered are formulated as *maxdet*-problems subject to LMI constraints. This is convenient when it comes to introducing several constraints on the orientation. A simple example when the directions of the x-axis and z-axis of the end-effector frame are specified independently is shown.

9.3 Representing Rotations

Most of the fundamental principles of rotation were presented in two papers by Leonhard Euler in 1775 (Alpern et al., 1993). The first paper shows that any rotation can be accomplished by a sequence of three rotations about the coordinate axes. In the second paper, Euler states that any orientation can be represented by a rotation of some angle ϕ about some fixed axis $\boldsymbol{n}$. He also shows that the composition of two rotations is itself a rotation.

9.3.1 The Unit Quaternion

The unit quaternion representation closely relates to the results presented in Euler's second paper. A good introduction to quaternions is found in Kuipers (2002). Any positive rotation ϕ about a fixed unit vector $\boldsymbol{n}$ can be represented by the four-tuple

$$Q = \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix}, \tag{9.1}$$

where $q_0 \in \mathbb{R}$ is known as the scalar part and $\boldsymbol{q} \in \mathbb{R}^3$ as the vector part. $Q(\phi, \boldsymbol{n})$ is written in terms of ϕ and $\boldsymbol{n}$ by

$$q_0 = \cos(\frac{\phi}{2}), \qquad \boldsymbol{q} = \sin(\frac{\phi}{2})\boldsymbol{n}. \tag{9.2}$$

Q is a quaternion of unit length and denoted a *unit quaternion*. Henceforth, all quaternions have unit length if not other is stated. Let $Q_P = \begin{bmatrix} p_0 & \boldsymbol{p}^\mathsf{T} \end{bmatrix}^\mathsf{T}$. The quaternion product of a rotation Q followed by a rotation Q_P is written in vector algebra notations as

$$Q_P * Q = \begin{bmatrix} p_0 q_0 - \boldsymbol{p} \cdot \boldsymbol{q} \\ p_0 \boldsymbol{q} + q_0 \boldsymbol{p} + \boldsymbol{p} \times \boldsymbol{q} \end{bmatrix}. \tag{9.3}$$

The cross product implies that quaternion multiplication is not commutative, as expected. Let $Q_P = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix}^\mathsf{T}$ and $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$. Then the quaternion product is written as

$$Q_P * Q = \begin{bmatrix} p_0 q_0 - p_1 q_1 - p_2 q_2 - p_3 q_3 \\ p_0 q_1 + p_1 q_0 + p_2 q_3 - p_3 q_2 \\ p_0 q_2 + p_2 q_0 + p_3 q_1 - p_1 q_3 \\ p_0 q_3 + p_3 q_0 + p_1 q_2 - p_2 q_1 \end{bmatrix}. \tag{9.4}$$

The quaternion product of two unit quaternions is a unit quaternion. By the definition of the quaternion the quaternions Q and $-Q$ produce the same rotation. This is referred to as the dual covering. The quaternion identity is given by $Q_I = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$.

A pure quaternion is a quaternion with zero scalar part. Any vector, $\bar{\boldsymbol{v}} = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T}$ can be represented by a pure quaternion

$$\boldsymbol{v} = \begin{bmatrix} 0 \\ \bar{\boldsymbol{v}} \end{bmatrix}. \tag{9.5}$$

The conjugate of a quaternion is defined as

$$Q^* = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{bmatrix}^\mathsf{T}. \tag{9.6}$$

9.3.2 Quaternions and Rotations

Let a vector, $\bar{\boldsymbol{v}}_1$, be represented by the pure quaternion $\boldsymbol{v}_1$. This vector can be rotated ϕ radians around the axis $\boldsymbol{n}$ by

$$\boldsymbol{v}_2 = Q * \boldsymbol{v}_1 * Q^*. \tag{9.7}$$

Every vector $\bar{\boldsymbol{v}} \in \mathbb{R}^3$ can be represented by a pure quaternion, hence $\boldsymbol{v}$ is not necessarily a unit quaternion. The quaternion, $Q(\phi, \boldsymbol{n})$, however, is unitary. This represents the angle and the axis that the vector $\bar{\boldsymbol{v}}_1$ is to be rotated about. The resulting vector, $\bar{\boldsymbol{v}}_2$, is then of the same length as $\bar{\boldsymbol{v}}_1$ if and only if Q is a unit quaternion. The quaternion representation also leads to a useful formula for finding the shortest rotation from one orientation to another. Let Q_P and Q be two orientations. Then, by taking

$$E = Q_P^* * Q, \tag{9.8}$$

E will rotate Q_P into Q by the shortest rotation.

Note that Equation (9.8) rotates one frame into another frame. By a *frame* it is meant a coordinate system in $\mathbb{R}^3$ using Cartesian coordinates. One frame with respect to another frame represents three degrees of freedom and is referred to as *orientation*. The inertial frame is denoted, $\mathcal{F}_I$ and the frame that correspond to the inertial frame by a rotation Q from the inertial frame is denoted $\mathcal{F}_Q$. Equation (9.7) rotates one vector into another vector and has two degrees of freedom (e.g. longitude and latitude) (Ahuactzin and Gupka, 1999). A unit vector with respect to a unit reference vector is referred to as *direction*. Henceforth, the main concern is with the direction of the central axis, which is assumed to be the body frame z-axis of the end effector.

9.3.3 Rotation Sequences

In this paper, the orientation is represented by a rotation sequence of three rotations about the unitary axes. The ZYZ-sequence is given by first a rotation α about the z-axis followed by a rotation β about the new y-axis. This describes the direction of the z-axis. The last degree of freedom is given by the rotation γ about the z-axis. When the sequence is given, a one-to-one[1] mapping between (α, β, γ) and the quaternion $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ can be found whenever $\beta \neq 0$.

Given a quaternion Q. Then α, β and γ from the ZYZ-sequence are found by (From and Gravdahl, 2007b)

$$\alpha = \arctan2\left(\frac{q_2q_3 - q_0q_1}{q_0q_2 + q_1q_3}\right), \tag{9.9}$$

$$\beta = 2\arcsin\sqrt{q_1^2 + q_2^2}, \tag{9.10}$$

$$\gamma = \arctan2\left(\frac{q_2q_3 + q_0q_1}{q_0q_2 - q_1q_3}\right). \tag{9.11}$$

The following relations are also used in the following:

$$\alpha = \arctan(\frac{q_3}{q_0}) - \arctan(\frac{q_1}{q_2}), \tag{9.12}$$

$$\gamma = \arctan(\frac{q_3}{q_0}) + \arctan(\frac{q_1}{q_2}), \tag{9.13}$$

and hence

$$\alpha + \gamma = 2\arctan(\frac{q_3}{q_0}). \tag{9.14}$$

[1] If the dual covering of the quaternion is taken into account, a one-to-two mapping can be found.

The following lemmas will be used in the following to determine positive definiteness of a matrix.

Lemma 9.1. *(Sylvester's criterion) A matrix M is positive definite if and only if all of the leading principal minors are positive. M is positive semi-definite if all the leading principal minors are non-negative.*

Lemma 9.2. *A block diagonal matrix*

$$P = \begin{bmatrix} P_1 & 0 & \dots & 0 \\ 0 & P_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & P_k \end{bmatrix} \tag{9.15}$$

is symmetric (semi) positive definite if and only if each block $P_i, \quad i = 1, \dots, k$ is symmetric (semi) positive definite.

9.4 Orientation Error Constraints as LMIs

9.4.1 Cone

Assume that one would like to restrict the z-axis of $\mathcal{F}_Q$ to point in approximately the same direction as the z-axis of the inertial frame $\mathcal{F}_I$. This can be visualised by a cone of directions and restricted by $|\beta| \leq \beta_{lim}$ where $0 \leq \beta_{lim} \leq \pi$. The orientation error β can be found only from q_1 and q_2 from the quaternion Q by (9.10)

$$\beta = 2 \arcsin \sqrt{q_1^2 + q_2^2}. \tag{9.16}$$

Due to this observation, a test to verify if the z-axis of $\mathcal{F}_Q$ does not deviate from the z-axis of $\mathcal{F}_I$ by more than β_{lim} is given in the following.

Proposition 9.1. *Given a restriction in the orientation error, β_{lim}. Then the z-axis of $\mathcal{F}_Q$ rotated by $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ from the inertial frame $\mathcal{F}_I$ lies within the restrictions given by β_{lim} if and only if*

$$P = \begin{bmatrix} \eta & 0 & q_1 \\ 0 & \eta & q_2 \\ q_1 & q_2 & \eta \end{bmatrix} \succeq 0 \tag{9.17}$$

where $\eta = \sin \frac{\beta_{lim}}{2}$, $0 \leq \beta_{lim} \leq \pi$ and $\succeq$ means positive semi-definiteness of the symmetric matrix P.

Proof. As $\eta \geq 0$ and $\eta^2 \geq 0$, from Lemma 9.1 it only remains to test for $\det(P)$. The determinant of P is given by

$$\det(P) = \eta(\eta^2 - q_1^2 - q_2^2). \tag{9.18}$$

Note that $0 \leq \beta_{lim} < \pi \Rightarrow \eta \geq 0$ so that $\det(P) \geq 0$ can be written as

$$\begin{aligned} \eta^2 - q_1^2 - q_2^2 &\geq 0 \\ \eta^2 &\geq q_1^2 + q_2^2 \\ \eta &\geq \sqrt{q_1^2 + q_2^2} \\ \sin\frac{\beta_{lim}}{2} &\geq \sqrt{q_1^2 + q_2^2} \end{aligned} \tag{9.19}$$

As $0 \leq \sqrt{q_1^2 + q_2^2} \leq 1 \Rightarrow 0 \leq \arcsin\sqrt{q_1^2 + q_2^2}$, the following holds

$$0 \leq 2\arcsin\sqrt{q_1^2 + q_2^2} \leq \beta_{lim}. \tag{9.20}$$

Then Equation (9.16) concludes the proof as

$$0 \leq \beta \leq \beta_{lim}. \tag{9.21}$$

Similarly for strictly positive definiteness. □

Note that the restrictions in Proposition 9.1 are on the directions of the z-axis only and that rotations about the z-axis itself are not restricted (the pointing task). Note also that P is a symmetric matrix. This is an important property that will be used in the next sections.

9.4.2 Restriction on the Orientation about the Central Axis

In the following a condition on the orientation error about the central axis is given. Assume that the x-axis points in the direction of the velocity and that it is desired that the body frame x-axis points in approximately the direction of the x-axis of the reference orientation. Again consider the ZYZ-sequence. In the case when no orientation error is allowed for the direction of the central axis, this is given trivially by $|\gamma| < c_{max}$, where c_{max} is the maximum allowed orientation error of the x-axis. For the ZYZ-sequence the direction of the x-axis is given by both α, β and γ. Assume that the orientation error of the direction of the z-axis is restricted as in the previous section. When this is restricted to be relatively small, the error in the direction of the x-axis can be approximated by the error in the orientation about the central axis. This error is written as

$$\epsilon = \alpha + \gamma. \tag{9.22}$$

This leads to the following result.

Proposition 9.2. *Assume that the orientation error of the direction of the z-axis is small. Given a restriction in the orientation error ϵ_{lim} about the central axis, the x-axis of $\mathcal{F}_Q$ rotated by $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ from the inertial frame $\mathcal{F}_I$ lies within the restrictions given by ϵ_{lim} if and only if*

$$P = \begin{bmatrix} \kappa & \frac{q_3}{q_0} \\ \frac{q_3}{q_0} & \kappa \end{bmatrix} \succeq 0 \tag{9.23}$$

where $\kappa = \tan\frac{\epsilon_{lim}}{2}$.

Proof. The determinant of P is given by

$$\det(P) = \kappa^2 - \frac{q_3^2}{q_0^2} \tag{9.24}$$

As $0 \leq \tan\frac{\epsilon_{lim}}{2}$ for $0 \leq \epsilon_{lim} \leq \pi$, $\det(P) \geq 0$ is written as

$$\begin{aligned} \kappa^2 &\geq \frac{q_3^2}{q_0^2} \\ \kappa &\geq \left|\frac{q_3}{q_0}\right| \\ \tan\frac{\epsilon_{lim}}{2} &\geq \left|\frac{q_3}{q_0}\right| \\ \epsilon_{lim} &\geq |2\arctan(\frac{q_3}{q_0})| \end{aligned} \tag{9.25}$$

Then Equation (9.14) concludes that

$$\epsilon_{lim} \geq |\epsilon| \tag{9.26}$$

□

where ϵ is given by Equation (9.14).

9.4.3 Direction of the x-axis

Alternatively, one might want to restrict the direction of the x-axis directly. Note that the matrix given in the previous section is not affine and cannot be used directly as an LMI in a *maxdet*-problem. Hence, another matrix that is both symmetric and affine is proposed in the following.

Assume that the direction of the x-axis is to be restricted. Similarly to Equation (9.17), the requirement that the body frame x-axis is to point in the direction of the inertial frame x-axis is given by

$$P_2 = \begin{bmatrix} \xi & 0 & q_2 \\ 0 & \xi & q_3 \\ q_2 & q_3 & \xi \end{bmatrix} \succeq 0 \tag{9.27}$$

where $\xi = \sin\frac{\beta_{lim}}{2}$.

Also note that the results presented are not restricted to the global reference frame $\mathcal{F}_I$. Assume that the direction of the body frame x-axis is to point in an arbitrary direction given by the direction of the x-axis of $Q_d = \begin{bmatrix} d_0 & d_1 & d_2 & d_3 \end{bmatrix}^\mathsf{T}$. In order to apply the restriction given by (9.27), but to the direction of the x-axis of $\mathcal{F}_{Q_d}$ and not that of $\mathcal{F}_I$, Q is transformed back into the inertial frame and the test is performed on the transformed quaternion

$$Q_t = Q_d^* * Q = \begin{bmatrix} * \\ * \\ -d_2q_0 + d_0q_2 - d_3q_1 + d_1q_3 \\ -d_3q_0 + d_0q_3 - d_1q_2 + d_2q_1 \end{bmatrix} \tag{9.28}$$

Note that when Q_t is substituted into (9.27), P_2 is still symmetric and affine in Q.

9.4.4 Pyramid

Assume instead that one would like to restrict the allowed rotation differently around different axes. For example, if the set of allowed orientations is given by restrictions on the rotation about the x-axis followed by a rotation about the y-axis, this will result in a pyramid-shaped set of allowed directions. The following observations are important in the following.

Rotating the vector $\bar{\boldsymbol{v}}_1 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^\mathsf{T}$ by α about the x-axis of the inertial frame followed by a rotation β about the y-axis, also of the inertial frame, is given by

$$\bar{\boldsymbol{v}}_I = \begin{bmatrix}\cos\alpha\sin\beta \\ -\sin\alpha \\ \cos\alpha\cos\beta\end{bmatrix}. \tag{9.29}$$

For a rotation α about the x-axis of the inertial frame followed by a rotation β about the y-axis of the rotated coordinate system is given by

$$\bar{\boldsymbol{v}}_R = \begin{bmatrix}\sin\beta \\ -\sin\alpha\cos\beta \\ \cos\alpha\sin\beta\end{bmatrix}. \tag{9.30}$$

This can also be written as a quaternion Q. Let the vector $\bar{\boldsymbol{v}}_1 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^\mathsf{T}$, represent the z-axis, be rotated by Q into $\boldsymbol{v}_2 = Q * \boldsymbol{v}_1 * Q^*$. Then $\boldsymbol{v}_2$ is written as

$$\boldsymbol{v}_2 = \begin{bmatrix}0 \\ 2(q_0q_2 + q_1q_3) \\ 2(q_2q_3 - q_0q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2\end{bmatrix} \tag{9.31}$$

This is a point on the unit sphere.

Proposition 9.3. *Given a restriction α_{lim} in the orientation error about the x-axis of the inertial frame and β_{lim} in the orientation error about the y-axis of the rotated coordinate frame. Then the z-axis of frame $\mathcal{F}_Q$ rotated by the quaternion $Q = \begin{bmatrix}q_0 & q_1 & q_2 & q_3\end{bmatrix}^\mathsf{T}$ with respect to the inertial frame $\mathcal{F}_I$ lies within the restrictions given by β_{lim} if and only if*

$$P_1 = \begin{bmatrix}\eta & 0 & q_1 \\ 0 & \eta & q_0 \\ q_3 & q_2 & \eta\end{bmatrix} \geq 0 \tag{9.32}$$

where $\eta = \sqrt{\frac{\sin\beta_{lim}}{2}}$ and $\geq$ means positive semi-definiteness for the non-symmetric matrix P_1.

Proof. The determinant of P_1 is given by

$$\det(P_1) = \eta(\eta^2 - q_0q_2 - q_1q_3). \tag{9.33}$$

Assume $\det(P_1) \geq 0$

$$\begin{aligned}\eta^2 - q_0q_2 - q_1q_3 &\geq 0 \\ \eta^2 &\geq q_0q_2 + q_1q_3\end{aligned} \tag{9.34}$$

As $\beta_{lim} \geq 0$, comparing Equations (9.30) and (9.31) gives

$$\sin \beta_{lim} \geq 2|q_0 q_2 + q_1 q_3|. \tag{9.35}$$

Let

$$\beta' = \arcsin\left(2(q_0 q_2 + q_1 q_3)\right), \tag{9.36}$$

and the initial requirement is obtained by

$$-\beta_{lim} \leq \beta' \leq \beta_{lim} \tag{9.37}$$

where β' is the angle between the new z-axis and the yz- plane. □

Proposition 9.4. *Given a restriction α_{lim} in the orientation error about the x-axis and β_{lim} in the orientation error about the y-axis, both in the inertial frame. Then the z-axis of frame $\mathcal{F}_Q$ rotated by the quaternion $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ with respect to the inertial frame $\mathcal{F}_I$ lies within the restrictions given by α_{lim} if and only if*

$$P_2 = \begin{bmatrix} \xi & q_2 q_3 & 0 \\ q_2 q_3 & \xi & q_0 q_1 \\ 2\xi & q_0 q_1 & \xi \end{bmatrix} \geq 0 \tag{9.38}$$

where $\xi = \frac{\sin \alpha_{lim}}{2}$.

Proof. The determinant of P_2 is given by

$$\det(P_2) = \xi(\xi^2 - (q_0 q_1)^2 - (q_2 q_3)^2 + 2 q_0 q_1 q_2 q_3). \tag{9.39}$$

Assume $\det(P_1) \geq 0$

$$\begin{gathered} \xi^2 - (q_0 q_1)^2 - (q_2 q_3)^2 + 2 q_0 q_1 q_2 q_3 \geq 0 \\ \xi^2 \geq (q_0 q_1)^2 + (q_2 q_3)^2 - 2 q_0 q_1 q_2 q_3 \\ \xi^2 \geq (q_0 q_1 - q_2 q_3)^2 \\ |\xi| \geq |q_0 q_1 - q_2 q_3| \end{gathered} \tag{9.40}$$

As $\alpha_{lim} \geq 0$, comparing Equations (9.31) and (9.29) gives

$$\sin \alpha_{lim} \geq 2|q_0 q_1 - q_2 q_3|. \tag{9.41}$$

Let

$$\alpha = \arcsin\left(2(q_0 q_1 - q_2 q_3)\right), \tag{9.42}$$

and the initial requirement is obtained by

$$-\alpha_{lim} \leq \alpha \leq \alpha_{lim}. \tag{9.43}$$

□

Note that in Proposition 9.3 the second rotation is with respect to the rotated coordinate frame and restricts only the rotations about the y-axis while in Proposition 9.4 the second rotation is with respect to the rotated coordinate frame and restricts the allowed rotations about the x-axis only. This simplifies the computations substantially. For small α and β this is a good approximation. The next step is to put the two restrictions together to one block-diagonal matrix. This shows how two restrictions on the orientation can be put together and represented as one constraint.

Note that the matrices given in Propositions 9.3 and 9.4 are not symmetric and that P_2 in (9.38) is not affine. Hence, the constraints cannot be represented as LMIs. They can, however, be represented as barrier functions given as the negative logarithm of the determinant. This is discussed shortly in Section 9.6.

Example 10. Given a restriction α_{lim} in the orientation error about the x-axis and β_{lim} in the orientation error about the y-axis. Then the z-axis of frame $\mathcal{F}_Q$ rotated by the quaternion $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ with respect to the inertial frame $\mathcal{F}_Q$ lies within the restrictions given by α_{lim} and β_{lim} if and only if

$$P = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} \geq 0 \tag{9.44}$$

where P_1 and P_2 are given as in Equations (9.32) and (9.38).

Alternatively, if one would like to restrict the orientation about the x-axis followed by the orientation about the new y-axis, this can be achieved by substituting

$$\eta = \sqrt{\frac{\sin \alpha_{lim} \cos \beta}{2}} \tag{9.45}$$

where $\beta = \arcsin 2(q_0 q_2 + q_1 q_3)$ for η into Equation (9.32) which will give the exact solution.

9.5 Applications

In this section it is shown how the results from the previous section can be used as LMIs in an optimisation problem.

9.5.1 Analytic Centering

The problem

$$\begin{array}{ll} \text{minimise} & \phi(x) = \log \det G(x)^{-1} \\ \text{subject to} & G(x) \succ 0 \end{array} \tag{9.46}$$

where

$$G(x) = G_0 + x_1 G_1 + x_2 G_2 + \cdots + x_m G_m, \tag{9.47}$$

is known as the analytic centering problem. If the feasible set $\boldsymbol{X} = \{x | G(x) \succ 0\}$ is non-empty and bounded, then the matrices $G_i, i = 1, ..., m$ are linearly independent and the objective function is strictly convex on $\boldsymbol{X}$ (Vandenberghe et al., 1996). In this case, it

can be guaranteed that the optimality condition $\nabla\phi(x^*) = 0$, for an optimal solution x^*, can be reached.

In our case, for the specification of the z-axis:

$$\begin{array}{ll} \text{minimise} & \phi(x) = \log\det P(x)^{-1} \\ \text{subject to} & P(x) \succ 0 \end{array} \tag{9.48}$$

where P is given by Equation (9.17) and can be written as

$$P(x) = P_0 + x_1P_1 + x_2P_2 + x_3P_3 + x_4P_4, \tag{9.49}$$

where
$$x_1 = q_0, \quad x_2 = q_1, \quad x_3 = q_2, \quad x_4 = q_3, \tag{9.50}$$

$$P_0 = \begin{bmatrix} \eta & 0 & 0 \\ 0 & \eta & 0 \\ 0 & 0 & \eta \end{bmatrix}, \qquad P_1 = P_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$P_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \qquad P_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

q_0 and q_3 do not affect the solution and can be eliminated from the equations.

The optimal solution to this problem is given by the set

$$Q_{opt} = \{Q \mid q_1 = q_2 = 0\} \tag{9.51}$$

or similarly by all quaternions on the form

$$Q_{opt} = \begin{bmatrix} \cos\frac{\psi}{2} & 0 & 0 & \sin\frac{\psi}{2} \end{bmatrix}^\mathsf{T}. \tag{9.52}$$

The solution to this problem is trivially given by all the orientations that make the body frame z-axis point in the direction of the inertial frame z-axis. We now look into the case when an additional constraint is added. Two cases are considered; (a) two constraints are represented by the positive definiteness of two matrices P_1 and P_2 and the determinant of the block diagonal matrix $P = \text{Blockdiag}(P_1, P_2)$ is minimised; and (b) P_1 is minimised and P_2 is a constraint.

9.5.2 Blockdiagonal G-matrix

To combine the restrictions of the x- and z-axes, substitute (9.28) into (9.27), denote the resulting matrix F, and write it on the form of (9.49) so that

$$F(x) = F_0 + x_1F_1 + x_2F_2 + x_3F_3 + x_4F_4, \tag{9.53}$$

$$x_1 = q_0, \quad x_2 = q_1, \quad x_3 = q_2, \quad x_4 = q_3, \tag{9.54}$$

$$F_0 = \begin{bmatrix} \xi & 0 & 0 \\ 0 & \xi & 0 \\ 0 & 0 & \xi \end{bmatrix}, \qquad F_1 = \begin{bmatrix} 0 & 0 & -d_2 \\ 0 & 0 & -d_3 \\ -d_2 & -d_3 & 0 \end{bmatrix},$$

$$F_2 = \begin{bmatrix} 0 & 0 & -d_3 \\ 0 & 0 & d_2 \\ -d_3 & d_2 & 0 \end{bmatrix}, \quad F_3 = \begin{bmatrix} 0 & 0 & d_0 \\ 0 & 0 & -d_1 \\ d_0 & -d_1 & 0 \end{bmatrix},$$

$$F_4 = \begin{bmatrix} 0 & 0 & d_1 \\ 0 & 0 & d_0 \\ d_1 & d_0 & 0 \end{bmatrix}.$$

The problem can now be formulated as follows:

$$\begin{array}{ll} \text{minimise} & \phi(x) = \log\det \begin{bmatrix} P(x) & 0 \\ 0 & F(x) \end{bmatrix}^{-1} \\ \text{subject to} & \begin{bmatrix} P(x) & 0 \\ 0 & F(x) \end{bmatrix} \succ 0 \end{array} \tag{9.55}$$

for which the solution is the orientation which minimises the error both of the x-axis and the z-axis with a "metric" that increases exponentially with the shortest angular distance from the desired directions of the x- and z-axes.

9.5.3 LMI Constraint

Alternatively the determinant of P can be minimised under the constraint $F(x) \succeq 0$.

$$\begin{array}{ll} \text{minimise} & \phi(x) = \log\det P(x)^{-1} \\ \text{subject to} & P(x) \succ 0 \\ & F(x) \succeq 0. \end{array} \tag{9.56}$$

Hence, the optimal solution of the direction of the z-axis is found and the direction of the x-axis is within the restrictions. If the x-axis is close or far from the desired direction does not affect the solution.

9.5.4 Normalisation

The optimisation algorithms described optimise freely over all quaternions, and it is thus not guaranteed, nor likely, that the resulting quaternion is of unit length. One simple, though not very mathematically sound solution is to optimise freely over all quaternions and then normalise the result afterwards. Another option is to add the constraint $|Q| = 1$ in the optimisation algorithm which guarantees that the search space is only the set of quaternions of unit length. For the restriction given by Equation (9.17), it is for example sufficient to add the restriction

$$q_1^2 + q_2^2 \leq 1. \tag{9.57}$$

9.6 Future Work

The examples shown in this paper are all very basic. They do, however, show how the formulation allows the programmer to include the constraints on the orientation error in optimisation problems in the form of linear matrix inequalities. Constraints in the form of

LMIs are convenient in the sense that they are easily combined with other LMIs into one "big" LMI. This is exploited in for example Han et al. (2000).

Consider the LMI

$$\text{Blockdiag}(P_1(x), P_2(x), \ldots, P_k(x)) \succeq 0. \tag{9.58}$$

Let $P_1(x_1)$ be the constraint on the orientation error. Then, $P_2(x) \ldots P_k(x)$ may impose other constraints on the optimisation problem. These may be related to the orientation, as in (9.55). The true advantage of representing the constraints as LMIs, however, only come apparent when the constraints on the orientation is combined with other constraints such as joint torque or the quality of the job performed. This is left as future research topics.

In this paper only LMIs have been considered, but the restrictions on the orientation could just as well have been formulated as for example barrier functions. Given the restrictions in Proposition 9.1, this can be formulated as a the barrier for the cone constraint

$$\phi = -\log(\eta^2 - q_1^2 - q_2^2) \tag{9.59}$$

which is then included in the barrier subproblem

$$\begin{array}{ll} \text{minimise} & F(x) + \phi(x) \\ \text{subject to} & P(x) = 0 \end{array} \tag{9.60}$$

where $F(x)$ is the objective function that we want to minimise and $P(x)$ is some additional constraint.

9.7 Conclusions

This paper casts constraints on the orientation error into linear matrix inequalities. For many practically important examples, it has been shown that the continuous set of orientations that satisfy a given constraint can be represented by a positive definiteness constraint of a certain matrix where the four real quaternion-quantities q_0, q_1, q_2 and q_3 are the entries. Some simple examples of how to include the LMIs into the *maxdet*-problem are given. The LMIs can be included in optimisation algorithms to find the optimal orientation of some devise, such as a satellite or the end effector of a robotic manipulator, subject to some optimisation criteria.

Chapter 10

A Real-Time Algorithm to Determine the Optimal Paint Gun Orientation in Spray Paint Applications

10.1 Abstract

In this paper we present a method for increasing the speed at which a standard industrial manipulator can paint a surface. The approach is based on the observation that a small error in the orientation of the end effector does not affect the quality of the paint job. It is far more important to maintain constant velocity throughout the trajectory. We consider the freedom in the end-effector orientation as functional redundancy and represent the restriction on the orientation error as barrier functions or linear matrix inequalities. In doing this we cast the problem of finding the optimal orientation at every time step into a convex optimisation problem that can be solved very efficiently and in real time. We show that the approach allows the end effector to maintain a higher constant velocity throughout the trajectory guaranteeing uniform paint coating and substantially reducing the time needed to paint the object.

***Note to Practitioners*—This paper is motivated by the observation that uniform paint coating cannot be achieved in steep turns. Even if the manipulator possesses the necessary actuator torques to maintain constant speed for a straight line trajectory the torques needed to maintain constant velocity during turn are far higher. Thus, the operator has to lower the trajectory speed, also in the straight line segments where this would normally not be necessary, or accept a thicker layer of paint in the turns. The method proposed in this paper is to implement a slightly different planning algorithm in turns allowing the paint gun to follow the trajectory with a higher constant velocity. This will allow the paint gun to follow the trajectory, including**

both straight line segments and turns, with constant velocity and thus achieve uniform paint coating. We show how to choose the desired orientation of the paint gun at every time step and present the explicit expressions for solving and implementing the algorithms.

The approach can also be used for other applications where introducing a freedom in the end-effector orientation improves performance, such as welding and high-pressure water blasting.

Keywords— Spray painting, assembly-line manufacturing, convex optimisation, robotics, functional redundancy, modelling.

10.2 Introduction

One of the most important benefits of introducing industrial manipulators to the assembly line automotive manufacturing in the 1980s was the removal of all human workers from the spray paint area, relieving them from a toxic working environment. It is crucial for the flow of the automotive assembly line that the spray painting is performed both with high quality and in an efficient manner. In this paper we address the problem of reducing the time needed to paint a surface without compromising the quality of the coating. This is based on the observation that the velocity at which the end effector follows the path is far more important to guarantee uniform paint coating than the orientation of the end effector.

We assume that the tool centre point trajectory, i.e. the trajectory at the surface that the paint gun is to follow, is known. Several approaches for finding the optimal path in terms of speed, coverage and paint waste have been presented in literature. An automatic trajectory planning system is presented in Suh and I.-K. Woo (1991). Both the painting mechanics and the robot dynamics are used to find the optimal trajectory with respect to paint uniformity and cycle time given a CAD model. Ramabhadran and Antonio (1997); Antonio (1994) present a computationally efficient formulation of the trajectory tracking problem in spray paint application while Kim and Sarma (2003) find the optimal sweeping paths by minimising the cycle time subject to actuator speed limits and coating thickness.

Some work has also been done on modelling the paint composition on a surface. Hertling et al. (1996) present a mathematical model of the paint coating for a tilted gun and Conner et al. (2005) develop computationally tractable analytic deposition models that allow us to include the paint model, including the orientation with respect to the surface, when considering the paint coating. Smith et al. (2001) discuss the problem of minimising the orientation error when following curved surfaces and Atkar et al. (2005) include the paint model in their framework for optimising cycle time and coating quality.

In Potkonjak et al. (2000) the idea of introducing the paint quality as a constraint and minimise some additional cost function was presented. This opens for the possibility of allowing a small error in the orientation of the end effector in order to increase the velocity of the paint gun, reduce torques and so on. It was shown in From and Gravdahl (2007a) that by allowing an orientation error, the speed and quality of the job was improved. However, the optimal orientation error was chosen intuitively and the approach presented was not suitable for implementation in an optimisation algorithm.

In Buss et al. (1996) the problem of friction force limit constraints was transformed into a problem of testing for positive definiteness of a certain matrix. In From and Gravdahl

(2008d) the same ideas were used to convert the problem of orientation error constraints into a test of positive definiteness of a matrix. For different types of orientation errors, a suitable matrix was found and it was shown that positive definiteness of this matrix is equivalent to an orientation satisfying the given restrictions.

By transforming the non-linear orientation constraints into positive definiteness constraints imposed on certain symmetric matrices we transform the problem of finding the optimal orientation into an optimisation problem on the smooth manifold of linearly constrained positive definite matrices. For the special case of positive definite symmetric matrices, the constraints can be written on the form of linear matrix inequalities (LMIs). We also show how to write the constraints as barrier functions and how to solve these. Convex optimisation problems involving LMIs or barrier functions have been extensively studied in literature, and reliable and efficient solutions are known (see Vandenberghe et al. (1996); Boyd and Vandenberghe (2004); Boyd et al. (1994)).

10.3 Problem Statement

There are two main factors that play an important role in obtaining uniform paint coating in automotive manufacturing. The first is to move the paint gun with constant velocity throughout the trajectory. This is in general an easy task in following straight lines but can be a challenge in turns where high accelerations are required. The second factor is the orientation of the paint gun with respect to the surface, which should be orthogonal. It can be shown that the velocity of the paint gun is far more important than the orientation when it comes to uniform paint coating. A small orientation error ($<20^\circ$) in the paint gun does not affect the quality of the coating to the same extent as changes in the velocity. Based on these observations we represent the orientation not as one frame, but as a constrained continuous set of frames. The problem treated in this paper is then formulated as follows:

Given a maximum allowed orientation error of the paint gun and a trajectory on the surface that the paint gun is to follow with constant velocity and with a fixed distance from the paint gun to the trajectory. Then the problem is to find the orientation of the paint gun at every point on the trajectory that allows it to follow the trajectory with the highest possible constant velocity.

We note that in this paper we do not require the orientation to be optimal. The optimal solution to this problem, considering both kinematics and dynamics, is extremely complex. However, we formulate the problem as an optimisation problem based on a simple and intuitive cost function and show that the solution to this problem substantially improves performance. In the following we will denote the solution to this optimisation problem the "optimal orientation" although strictly speaking there might exist other orientations that improve performance even further.

We consider a standard industrial manipulator, in our case the ABB IRB-5400 series which is illustrated in Fig. 10.1. The first three joints are referred to as the main axes, or the main joints. These are the strongest joints and also the ones that require the most energy. While the main axes are mainly used for positioning the paint gun, the last three joints, referred to as the wrist joints, determine the orientation of the paint gun. We fix the inertial reference frame to the base of the manipulator. We also attach a frame to the end effector of the manipulator, in our case the paint gun. This is attached so that the

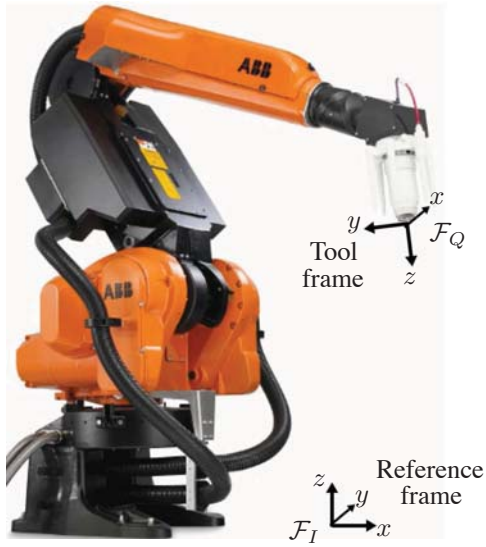


Figure 10.1: The ABB IRB spray paint robot with the definitions of the reference and tool frames. Picture courtesy of ABB Robotics.

end-effector z-axis is aligned with the direction of the paint flow. This axis is referred to as the central axis of the end effector.

To find the optimal orientation we first need to define a set of allowed orientations from which we can choose the optimal one. This set of orientations is defined using the unit quaternion which allows us to re-write the constraints using very simple expressions. Sections 10.4 and 10.5 give a brief background on representing orientations and continuous sets of orientations of rigid bodies. We also show how we can re-write restrictions on the direction of the central axis as a simple constraint on the unit quaternion. In Section 10.6 we present the theoretical background on how to write constraints on the orientation in a convex optimisation setting and in Section 10.7 we provide the equations needed for implementing the algorithms such that a solution can be found in real time.

In Section 10.8 we show how we can increase the speed at which the manipulator can paint a given surface without compromising the paint quality. The solution in itself is very simple. It basically allows us to distribute the work-load more evenly on the different joints. In our case we find that for the main joints the actuator torques are very close to the torque limits while the wrist joints use only a fraction of the torque available. We thus choose the orientations in a way that will make the main joints move less, and thus require less torque. One easy way to do this is to force the position of the wrist towards the centre of the surface reducing the length of its trajectory. Keeping in mind that the main joints are mainly used for displacement, this will reduce the required torques of these joints. Section 10.8 also includes several simulations to verify the efficiency of the approach presented.

10.4 Representing Rotations

10.4.1 The Unit Quaternion

The unit quaternion is well suited for representing orientations or continuous sets of orientations of rigid bodies. A good introduction to quaternions is found in Kuipers (2002). Any positive rotation ϕ about a fixed unit vector $\boldsymbol{n}$ can be represented by the four-tuple

$$Q = \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix}, \tag{10.1}$$

where $q_0 \in \mathbb{R}$ is known as the scalar part and $\boldsymbol{q} \in \mathbb{R}^3$ as the vector part. $Q(\phi, \boldsymbol{n})$ is written in terms of ϕ and $\boldsymbol{n}$ by

$$q_0 = \cos\left(\frac{\phi}{2}\right), \qquad \boldsymbol{q} = \sin\left(\frac{\phi}{2}\right)\boldsymbol{n}. \tag{10.2}$$

Q is a quaternion of unit length and denoted a *unit quaternion*. Henceforth, all quaternions have unit length if not other is stated. Let $Q_P = \begin{bmatrix} p_0 & \boldsymbol{p}^\mathsf{T} \end{bmatrix}^\mathsf{T}$. The quaternion product of a rotation Q followed by a rotation Q_P is written in vector algebra notations as

$$Q_P * Q = \begin{bmatrix} p_0 q_0 - \boldsymbol{p} \cdot \boldsymbol{q} \\ p_0 \boldsymbol{q} + q_0 \boldsymbol{p} + \boldsymbol{p} \times \boldsymbol{q} \end{bmatrix}. \tag{10.3}$$

The cross product implies that quaternion multiplication is not commutative, as expected. Let $Q_P = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix}^\mathsf{T}$ and $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$. Then the quaternion product is written as

$$Q_P * Q = \begin{bmatrix} p_0 q_0 - p_1 q_1 - p_2 q_2 - p_3 q_3 \\ p_0 q_1 + p_1 q_0 + p_2 q_3 - p_3 q_2 \\ p_0 q_2 + p_2 q_0 + p_3 q_1 - p_1 q_3 \\ p_0 q_3 + p_3 q_0 + p_1 q_2 - p_2 q_1 \end{bmatrix}. \tag{10.4}$$

The quaternion product of two unit quaternions is a unit quaternion. From the definition of the quaternion we see that the quaternions Q and $-Q$ produce the same rotation. This dual covering allows every rotation to be described twice. In this paper all angles are assumed to be in the interval $[-\pi, \pi]$ so every orientation corresponds to one specific quaternion. It is also assumed that all angles of inverse trigonometric functions are in this interval with the correct sign. For $\arctan$, this is denoted $\arctan 2$. The quaternion identity representing the inertial frame is given by $Q_I = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$.

A pure quaternion is a quaternion with zero scalar part. Any vector, $\bar{\boldsymbol{v}} = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T}$ can be represented by a pure quaternion $\boldsymbol{v} = \begin{bmatrix} 0 & \bar{\boldsymbol{v}}^\mathsf{T} \end{bmatrix}^\mathsf{T}$. Finally the conjugate of a quaternion is defined as $Q^* = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{bmatrix}^\mathsf{T}$.

10.4.2 Vector Rotations

Let a vector $\bar{\boldsymbol{v}}_1$ be represented by the pure quaternion $\boldsymbol{v}_1$. This vector can be rotated ϕ radians around the axis $\boldsymbol{n}$ by

$$\boldsymbol{v}_2 = Q * \boldsymbol{v}_1 * Q^*. \tag{10.5}$$

Every vector $\bar{\boldsymbol{v}} \in \mathbb{R}^3$ can be represented by a pure quaternion, hence $\boldsymbol{v}$ is not necessarily of unit length. The quaternion Q, however, is unitary. This represents the angle and the axis that the vector $\bar{\boldsymbol{v}}_1$ is rotated about. The resulting vector $\bar{\boldsymbol{v}}_2$ is then of the same length as $\bar{\boldsymbol{v}}_1$ if and only if Q is a unit quaternion.

Note that Equation (10.3) rotates one frame into another frame. By a *frame* it is meant a coordinate system in $\mathbb{R}^3$ using Cartesian coordinates. One frame with respect to another frame represents three degrees of freedom and is referred to as *orientation*. The reference frame is the inertial frame denoted $\mathcal{F}_I$ and the frame that corresponds to the inertial frame by a rotation Q is denoted $\mathcal{F}_Q$. Equation (10.5), however, rotates one vector into another vector and represents two degrees of freedom, i.e. a point on a sphere. A unit vector with respect to a unit reference vector is referred to as *direction*. Henceforth, the main concern is with the direction of the central axis, which is assumed to be the body frame z-axis of the end effector. We refer to van der Ha and Shuster (2009) for a good reference on vectors and attitudes. The following lemmas will also be used.

Lemma 10.1. *(Sylvester's criterion) A matrix P is positive definite if and only if all the leading principal minors are positive. P is positive semi definite if all the leading principal minors are non-negative.*

Lemma 10.2. *A block diagonal matrix $P = Blockdiag(P_1, \ldots, P_i, \ldots, P_k)$ is symmetric positive definite if and only if each block P_i, $i = 1, \ldots, k$ is symmetric positive definite. P is positive semi definite if each block is positive semi definite.*

10.5 Quaternion Volumes

We start by representing a continuous set of orientations defined by a set of constraints in Euler angles and a sequence of rotations. This allows us to find the corresponding constraints on the quaternion entries q_0, q_1, q_2 and q_3. We denote this continuous set of quaternions a *quaternion volume*. We then use this intuitive and well defined tool in the next sections to represent these constraints as LMIs or barrier functions.

10.5.1 General Definition

A set of frames corresponding to a reference frame by a rotation ϕ about a fixed axis $\boldsymbol{n}$ can be represented as

$$Q(\phi, \boldsymbol{n}), \qquad \text{for } \phi_{min} \leq \phi \leq \phi_{max}. \tag{10.6}$$

When the rotations are not limited to one axis only, a more general description of all allowed orientations can be represented by a sequence of rotations given by the quaternion product of two or more quaternions and their restrictions.

Definition 10.1 (Quaternion Volume)**.** A quaternion volume $Q^{\otimes}$ is defined as

$$\begin{aligned} Q^{\otimes} \triangleq \{Q(\phi_1, \ldots, \phi_n, \boldsymbol{n}_1, \ldots, \boldsymbol{n}_n) \mid \phi_{1,min} \leq \phi_1 \leq \phi_{1,max} \\ \vdots \\ \phi_{n,min} \leq \phi_n \leq \phi_{n,max}\} \end{aligned} \tag{10.7}$$

for $n \geq 1$ and where

$$Q(\phi_1, \ldots, \phi_n, \boldsymbol{n}_1, \ldots, \boldsymbol{n}_n) = Q(\phi_n, \boldsymbol{n}_n) * \cdots * Q(\phi_1, \boldsymbol{n}_1). \tag{10.8}$$

In this paper, only sets of frames that can be described by a sequence of rotations about the coordinate axes are treated. We refer to Hanson (2006) and From and Gravdahl (2007b) for a detailed discussion on quaternion volumes.

10.5.2 Reorientation of Quaternion Volumes

The quaternion product of two quaternion volumes, or a quaternion volume and a quaternion, is itself a quaternion volume. We can use this observation to transform quaternion volumes and to represent them in a rotated coordinate system.

Let $Q^{\otimes}$ be a quaternion volume and the quaternion P represent some transformation on $Q^{\otimes}$. Then the transformation $Q_P^{\otimes} = Q^{\otimes} * P$ rotates the entire set of frames by a rotation P. Similarly, the transformation $Q_P^{\otimes} = Q^{\otimes} * P^*$ allows the set of frames represented by the quaternion volume to be represented with respect to a new reference frame P. The transformation induced by changing from one reference frame to another is called *reorientation* (Alpern et al., 1993).

Proposition 10.1 (Transformation of Quaternion Volumes)**.** *Any quaternion volume $Q^{\otimes}$ represented with respect to the reference frame can be transformed into another quaternion volume by*

$$Q_P^{\otimes} = Q^{\otimes} * P, \tag{10.9}$$

where the orientations represented by $Q_P^{\otimes}$ relate to P in the same way as $Q^{\otimes}$ relates to the reference frame.

Proof. The quaternion product $E = Q * P$ can be viewed upon as a rotation P followed by a rotation Q with respect to the *new* frame. Hence, E relates to P in the same way as Q relates to the reference frame. By the same argumentation the quaternion volume $Q_P^{\otimes}$ relates to P in the same way as $Q^{\otimes}$ relates to the reference frame. □

In Proposition 10.1 the reference frame is kept constant and the quaternion volume is rotated by P. Reorientation, however, is a rotation of the reference frame (change of observer) while the quaternion volume is kept constant. The proof of the reorientation $Q_P^{\otimes} = Q^{\otimes} * P^*$ is constructed in the same way as the proof of Proposition 10.1.

10.5.3 The Pointing Task

We now show how to represent the freedom of the pointing task as a quaternion volume. First assume that the z-axis of the end effector must be aligned with the z-axis of $\mathcal{F}_I$. This gives the end effector one degree of rotational freedom about the z-axis. The pointing task can be represented by an arbitrary rotation ψ about the z-axis as the quaternion volume

$$Q_{pt}^{\otimes} = \begin{bmatrix} \cos\left(\frac{\psi}{2}\right) & 0 & 0 & \sin\left(\frac{\psi}{2}\right) \end{bmatrix}^{\mathsf{T}}, \quad \text{for} \quad -\pi < \psi \leq \pi. \tag{10.10}$$

The quaternion volume is thus given with respect to the reference frame. Assume the desired quaternion volume instead is to be rotated by $Q_d = \begin{bmatrix} d_0 & d_1 & d_2 & d_3 \end{bmatrix}^{\mathsf{T}}$ from the

reference frame. The quaternion volume that describes all orientations where the z-axis points in the same direction as the z-axis of Q_d is given by $Q_d^{\otimes} = Q_{pt}^{\otimes} * Q_d$ so that

$$Q_d^{\otimes} = \begin{bmatrix} d_0 \cos(\frac{\psi}{2}) - d_3 \sin(\frac{\psi}{2}) \\ d_1 \cos(\frac{\psi}{2}) - d_2 \sin(\frac{\psi}{2}) \\ d_2 \cos(\frac{\psi}{2}) + d_1 \sin(\frac{\psi}{2}) \\ d_3 \cos(\frac{\psi}{2}) + d_0 \sin(\frac{\psi}{2}) \end{bmatrix}, \text{ for } -\pi < \psi \leq \pi. \tag{10.11}$$

Example 11. If the desired orientation is chosen so that the z-axis of the end effector always points in the opposite direction of the z-axis of $\mathcal{F}_I$ by a rotation about the y-axis $Q_d = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^{\mathsf{T}}$, Equation (10.11) simplifies to

$$Q_d^{\otimes} = \begin{bmatrix} 0 & -\sin(\frac{\psi}{2}) & \cos(\frac{\psi}{2}) & 0 \end{bmatrix}^{\mathsf{T}}, \quad \text{for } -\pi < \psi \leq \pi. \tag{10.12}$$

All the quaternions that satisfy this restriction result in an end effector pointing in the opposite direction of the z-axis of $\mathcal{F}_I$. Wee see this by rotating the vector $\hat{\boldsymbol{v}}_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}}$ by $Q_d^{\otimes}$. Then for $-\pi < \psi \leq \pi$ we have

$$\boldsymbol{z}^{\otimes} = Q_d^{\otimes} * \boldsymbol{v}_z * (Q_d^{\otimes})^* \tag{10.13}$$

$$= \begin{bmatrix} 0 \\ -\sin(\frac{\psi}{2}) \\ \cos(\frac{\psi}{2}) \\ 0 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} * \begin{bmatrix} 0 \\ \sin(\frac{\psi}{2}) \\ -\cos(\frac{\psi}{2}) \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\frac{\psi}{2}) \sin(\frac{\psi}{2}) - \cos(\frac{\psi}{2}) \sin(\frac{\psi}{2}) \\ 0 \\ 0 \\ -\cos^2(\frac{\psi}{2}) - \sin^2(\frac{\psi}{2}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}. \tag{10.14}$$

10.5.4 Cone Shaped Quaternion Volumes by Rotations Sequences

A rotation sequence describes a rotation about one coordinate axis followed by a rotation about another coordinate axis in the rotated coordinate system. A general framework on how to construct easily visualisable quaternion volumes by rotation sequences is presented. We show how to construct different types of quaternion volumes and how these relate to the different rotation sequences. This will allow the programmer to choose the quaternion volume most appropriate for the task in hand or to define volumes using other rotation sequences to obtain a new shape well suited for a specific task. The rotation sequence starts with two subsequent rotations about two coordinate axes, represented by the quaternion Q_s. This defines the direction of the central axis, which is our main concern. The last degree of freedom is added by a rotation about the central axis itself, here the z-axis, by Q_z. Then the orientation of the end effector is described by

$$Q = Q_z * Q_s. \tag{10.15}$$

We will look into two different rotation sequences, the ZYZ-sequence and the XYZ-sequence. For the ZYZ-sequence the direction of the central axis is determined by a

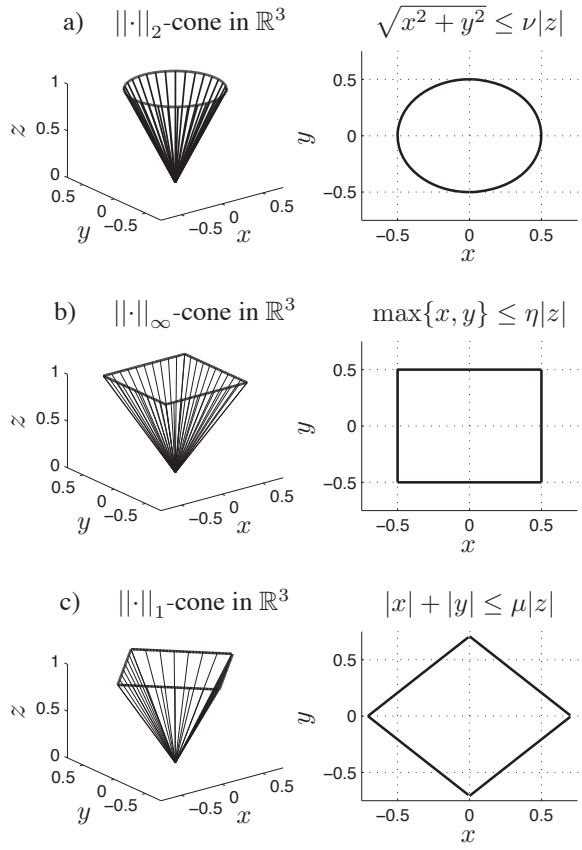


Figure 10.2: Different convex cones in $\mathbb{R}^3$. The cone defined by the 2-norm is self dual (setting $\mu = 1$). The cone defined by the ∞-norm is the dual of the cone defined by the 1-norm. The illustrations of the ∞- and 1-norms are good approximations for small rotations.

rotation about the z-axis followed by a rotation about the new y-axis. Thus, the difference in the direction between the new and the old central axis is given by the rotation about the y-axis only. For the XYZ-sequence, however, this difference is given by the first two rotations. For both sequences the last degree of freedom is given by a rotation about the central axis itself and does not change its direction. Finally, the quaternion volume is given by restricting the allowed rotations of each quaternion.

We use norms in $\mathbb{R}^3$ to define the directions of the central axis. We consider the three cones given in Fig. 10.2. The cones are defined by the degree of the norm, representing the shape of the cone, and by a parameter ξ representing the size of the cone by

$$||x_1, x_2, \ldots, x_{n-1}|| \leq \xi |x_n|. \tag{10.16}$$

We are mainly concerned with the z-axis, so in $\mathbb{R}^3$ we write

$$||x, y|| \leq \xi|z|. \tag{10.17}$$

Example 12. Given a $||\cdot||_2$-cone with the parameter ν restricting the direction of the z-axis, i.e.

$$\sqrt{x^2 + y^2} \leq \nu|z|. \tag{10.18}$$

Then the maximum rotation allowed by this cone is $\beta_{lim} = \arctan \nu$ around any axis in the xy-plane. This is obtained by the ZYZ-sequence and can be visualised in Fig. 10.2a).

Example 13. Given a $||\cdot||_\infty$-cone with the parameter η restricting the direction of the z-axis, i.e.

$$\max\{x, y\} \leq \eta|z|. \tag{10.19}$$

Then the maximum rotation allowed by this cone is $\beta_{lim} = \arctan \eta$ around the coordinate axes (x- and y-axes) and $\beta_{lim} = \arctan \sqrt{2}\eta$ around the axes $x = \pm y$. This is obtained by the XYZ-sequence and can be visualised in Fig. 10.2b) for small rotations.

Example 14. Given a $||\cdot||_1$-cone with the parameter μ restricting the direction of the z-axis, i.e.

$$|x| + |y| \leq \mu|z|. \tag{10.20}$$

Then the maximum rotation allowed by this cone is $\beta_{lim} = \arctan \mu$ around the coordinate axes (x- and y-axes) and $\beta_{lim} = \arctan \frac{\mu}{\sqrt{2}}$ around the axes $x = \pm y$. This is the dual of the $||\cdot||_\infty$-cone and is visualised in Fig. 10.2c) for small rotations.

We note that the results are valid for rotations around globally defined x- and y-axes while the XYZ-sequence rotates about the rotated coordinate axes. For the ∞- and 1-norms this is thus an approximation and only valid for small rotations.

We will represent the desired orientations as the continuous set of directions of the central axis as described by the cones and a free rotation about the central axis itself. This set can be composed by a rotation sequence of quaternion volumes. Two rotation sequences are discussed in detail, the ZYZ-sequence, also considered in Alpern et al. (1993) and Hanson (2006), and the XYZ-sequence.

2-cone

The ZYZ-sequence allows the desired orientation to be defined as a set of vectors that span out a $||\cdot||_2$-cone about the reference z-axis and all orientations about these vectors. Let $Q_s(\alpha, \beta) = Q(\beta, \boldsymbol{y}) * Q(\alpha, \boldsymbol{z})$ where $Q(\alpha, \boldsymbol{z}) = \begin{bmatrix}\cos(\frac{\alpha}{2}) & 0 & 0 & \sin(\frac{\alpha}{2})\end{bmatrix}^\mathsf{T}$ and $Q(\beta, \boldsymbol{y}) = \begin{bmatrix}\cos(\frac{\beta}{2}) & 0 & \sin(\frac{\beta}{2}) & 0\end{bmatrix}^\mathsf{T}$ so that

$$Q_s(\alpha, \beta) = \begin{bmatrix}\cos(\frac{\alpha}{2})\cos(\frac{\beta}{2}) \\ \sin(\frac{\alpha}{2})\sin(\frac{\beta}{2}) \\ \cos(\frac{\alpha}{2})\sin(\frac{\beta}{2}) \\ \sin(\frac{\alpha}{2})\cos(\frac{\beta}{2})\end{bmatrix}. \tag{10.21}$$

α represents the allowed orientations about the z-axis of the first rotation while β is the allowed orientation about the new y-axis. If α has no restrictions, β defines the size of a

cone with the z-axis at the centre, illustrated in Fig. 10.2a). We let γ restrict the orientation about the z-axis itself and the corresponding quaternion volume is then given by

$$Q_d^{\otimes} = Q_z^{\otimes} * Q_s^{\otimes} = \begin{bmatrix} \cos(\frac{\beta}{2})\cos(\frac{\gamma}{2}+\frac{\alpha}{2}) \\ \sin(\frac{\beta}{2})\sin(\frac{\alpha}{2}-\frac{\gamma}{2}) \\ \sin(\frac{\beta}{2})\cos(\frac{\gamma}{2}-\frac{\alpha}{2}) \\ \cos(\frac{\beta}{2})\sin(\frac{\gamma}{2}+\frac{\alpha}{2}) \end{bmatrix} \tag{10.22}$$

and the restrictions

$$-\alpha_{lim} \leq \alpha \leq \alpha_{lim} \tag{10.23}$$
$$0 \leq \beta \leq \beta_{lim} \tag{10.24}$$
$$-\gamma_{lim} \leq \gamma \leq \gamma_{lim} \tag{10.25}$$

Example 15. Assume that the central axis is to point in the opposite direction of the z-axis of $\mathcal{F}_I$. Further assume that a small error β_{lim} in the direction is allowed and no restrictions on the rotation about the z-axis. The set of frames describing these orientations is given by (10.22) and the restrictions

$$-\pi < \alpha \leq \pi \tag{10.26}$$
$$\pi \leq \beta \leq \pi + \beta_{lim} \tag{10.27}$$
$$-\pi < \gamma \leq \pi \tag{10.28}$$

We can also substitute $\beta \leftarrow \pi + \beta$ and $\alpha \leftarrow -\alpha$ into (10.22)

$$Q_d^{\otimes} = Q_z^{\otimes} * Q_s^{\otimes} = \begin{bmatrix} -\sin(\frac{\beta}{2})\cos(\frac{\gamma}{2}-\frac{\alpha}{2}) \\ \cos(\frac{\beta}{2})\sin(\frac{\gamma}{2}+\frac{\alpha}{2}) \\ \cos(\frac{\beta}{2})\cos(\frac{\gamma}{2}+\frac{\alpha}{2}) \\ -\sin(\frac{\beta}{2})\sin(\frac{\alpha}{2}-\frac{\gamma}{2}) \end{bmatrix} \tag{10.29}$$

and restrictions (10.23)-(10.25). Note that Equation (10.29) can also be obtained by rotating the quaternion volume in (10.22) by π radians about the y-axis, i.e. by Equation (10.9) with $P = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^{\mathsf{T}}$ and $Q^{\otimes} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^{\mathsf{T}}$ as in (10.22) so that $Q_d^{\otimes} = \begin{bmatrix} -q_2 & q_3 & q_0 & -q_1 \end{bmatrix}^{\mathsf{T}}$, which is the same as (10.29).

∞-cone

The XYZ-sequence defines the $||\cdot||_\infty$-cone, or a square cone of allowed directions where the allowed orientations about the x-axis and the (new) y-axis are restricted. This is a good estimation of restricting the orientation about the globally defined x- and y-axes whenever the angles are kept small. $Q_s(\alpha, \beta)$ is then given by

$$Q_s(\alpha, \beta) = \begin{bmatrix} \cos(\frac{\alpha}{2})\cos(\frac{\beta}{2}) \\ \sin(\frac{\alpha}{2})\cos(\frac{\beta}{2}) \\ \cos(\frac{\alpha}{2})\sin(\frac{\beta}{2}) \\ -\sin(\frac{\alpha}{2})\sin(\frac{\beta}{2}) \end{bmatrix}. \tag{10.30}$$

The orientation is given by the quaternion volume

$$Q_d^{\otimes} = Q_z^{\otimes} * Q_s^{\otimes} \tag{10.31}$$

and the restrictions

$$-\alpha_{lim} \leq \alpha \leq \alpha_{lim} \tag{10.32}$$
$$-\beta_{lim} \leq \beta \leq \beta_{lim} \tag{10.33}$$
$$-\gamma_{lim} \leq \gamma \leq \gamma_{lim} \tag{10.34}$$

10.5.5 Quaternion Volume Test

We now derive a test to verify if a quaternion lies inside the desired quaternion volume. We will in turn use this to transform these restrictions into constraints that can be handled directly in convex optimisation problems. Consider a quaternion volume defined by the ZYZ-sequence. We show how to use the analytic expression of the quaternion volume to find test to verify if a query quaternion $Q_{qry} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^{\mathsf{T}}$ is an element of the quaternion volume. Equation (10.22) gives

$$\begin{bmatrix} \cos(\frac{\beta}{2})\cos(\frac{\gamma}{2}+\frac{\alpha}{2}) \\ \sin(\frac{\beta}{2})\sin(\frac{\alpha}{2}-\frac{\gamma}{2}) \\ \sin(\frac{\beta}{2})\cos(\frac{\gamma}{2}-\frac{\alpha}{2}) \\ \cos(\frac{\beta}{2})\sin(\frac{\gamma}{2}+\frac{\alpha}{2}) \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \begin{matrix} (I) \\ (II) \\ (III) \\ (IV) \end{matrix} \tag{10.35}$$

Then, from the Appendix we get

$$\alpha = \arctan(\frac{q_3}{q_0}) + \arctan(\frac{q_1}{q_2}), \tag{10.36}$$
$$\beta = 2\arcsin\sqrt{q_1^2 + q_2^2}, \tag{10.37}$$
$$\gamma = \arctan(\frac{q_3}{q_0}) - \arctan(\frac{q_1}{q_2}), \tag{10.38}$$

which gives

$$\alpha + \gamma = 2\arctan(\frac{q_3}{q_0}). \tag{10.39}$$

An alternative formulation is given by (From and Gravdahl, 2007b)

$$\alpha = \arctan2\left(\frac{q_2q_3 + q_0q_1}{q_0q_2 - q_1q_3}\right), \tag{10.40}$$

$$\beta = 2\arccos\sqrt{q_0^2 + q_3^2}, \tag{10.41}$$

$$\gamma = \arctan2\left(\frac{q_2q_3 - q_0q_1}{q_0q_2 + q_1q_3}\right). \tag{10.42}$$

10.5.6 Transformed Quaternion Volumes

The easiest way to verify if a query quaternion lies inside a quaternion volume transformed by Equation (10.9) is to transform the query quaternion by the opposite transformation P so that both the quaternion volume and the query quaternion are presented in the reference frame. Hence, the two problems below are identical.

$$Q_{qry} \in Q^{\otimes} * P \quad ? \tag{10.43}$$

$$Q_{qry} * P^* \in Q^{\otimes} \quad ? \tag{10.44}$$

This operation is computationally demanding. In the special case when an analytical expression of the transformed quaternion volume is given, as in Equation (10.9), the orientation should be found by a set of parameters similar to the ones found in Equations (10.40)-(10.42). We can obtain this when the quaternion volume is on a simple form, for example as in Equation (10.29) where the quaternion volume is rotated 180° around the y-axis. Then the query quaternion may be tested against the restrictions in (10.23)-(10.25) directly. By following the mathematics of Equations (10.35)-(10.42), α, β and γ are found with respect to the coordinate system of $P = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^{\mathsf{T}}$ by

$$\alpha_P = \arctan2\left(\frac{q_0 q_1 + q_2 q_3}{q_0 q_2 - q_1 q_3}\right), \tag{10.45}$$

$$\beta_P = 2\arcsin\sqrt{q_0^2 + q_3^2}, \tag{10.46}$$

$$\gamma_P = \arctan2\left(\frac{q_0 q_1 - q_2 q_3}{q_0 q_2 + q_1 q_3}\right). \tag{10.47}$$

Hence, as expected we get $\beta_P = \beta - \pi$, $\alpha_P = \alpha$ and $\gamma_P = -\gamma$.

10.6 Restrictions on Orientation Error in a Convex Optimisation Setting

In this section we show how the formalism of quaternion volumes naturally leads to formulating restrictions on the orientation as LMIs and barrier functions.

10.6.1 2-norm

Assume that we would like to restrict the z-axis of $\mathcal{F}_Q$ to point in approximately the same direction as the z-axis of the reference frame $\mathcal{F}_I$. This can be visualised by a cone of directions restricted by $|\beta| \leq \beta_{lim}$ where $0 \leq \beta_{lim} \leq \pi$. The orientation error β can be found from q_1 and q_2 by (10.37), i.e.

$$\beta = 2\arcsin\sqrt{q_1^2 + q_2^2}. \tag{10.48}$$

A test to verify if the z-axis of $\mathcal{F}_Q$ does not deviate from the z-axis of $\mathcal{F}_I$ by more than β_{lim} is given in the following.

Proposition 10.2. *Given a maximum allowed deviation in the direction of the z-axis, represented by the rotation β_{lim}. Then the z-axis of $\mathcal{F}_Q$ rotated by $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ from the reference frame $\mathcal{F}_I$ lies within the $||\cdot||_2$-cone defined by β_{lim} if and only if*

$$P = \begin{bmatrix} \eta & 0 & q_1 \\ 0 & \eta & q_2 \\ q_1 & q_2 & \eta \end{bmatrix} \succeq 0 \tag{10.49}$$

where $\eta = \sin \frac{\beta_{lim}}{2}$, $0 \leq \beta_{lim} \leq \pi$ and $\succeq$ means positive semi-definiteness of the symmetric matrix P.

Proof. As $\eta \geq 0$ and $\eta^2 \geq 0$, from Lemma 10.1 we have that $P \succeq 0$ if $\det(P) \geq 0$. The determinant of P is given by

$$\det(P) = \eta(\eta^2 - q_1^2 - q_2^2). \tag{10.50}$$

Note that $0 \leq \beta_{lim} < \pi \Rightarrow \eta \geq 0$ so that $\det(P) \geq 0$ can be written as

$$\begin{aligned} \eta^2 - q_1^2 - q_2^2 &\geq 0 \\ \sin \frac{\beta_{lim}}{2} &\geq \sqrt{q_1^2 + q_2^2} \end{aligned} \tag{10.51}$$

As $0 \leq \sqrt{q_1^2 + q_2^2} \leq 1 \Rightarrow 0 \leq \arcsin \sqrt{q_1^2 + q_2^2}$, we have

$$0 \leq 2 \arcsin \sqrt{q_1^2 + q_2^2} \leq \beta_{lim}. \tag{10.52}$$

Then Equation (10.48) concludes the proof as

$$0 \leq \beta \leq \beta_{lim}. \tag{10.53}$$

□

Note that the restrictions in Proposition 10.2 are on the directions of the z-axis only and that rotations about the z-axis itself are not restricted (the pointing task). Note also that P is symmetric and affine in Q. This is an important property as it allows us to represent the constraints as LMIs. The following follows directly from Proposition 10.2 and allows us to formulate the restrictions as a barrier function.

Corollary 10.1. Given a maximum allowed deviation in the direction of the z-axis, represented by the rotation β_{lim} and let $\eta = \sin \frac{\beta_{lim}}{2}$. Then the barrier function

$$\phi = -\log(\eta^2 - q_1^2 - q_2^2) \tag{10.54}$$

increases exponentially to infinity as the orientation approaches the orientation limit forcing the z-axis of $\mathcal{F}_Q$ rotated by $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ from the reference frame $\mathcal{F}_I$ to lie within the restrictions given by β_{lim}.

The proof of Corollary 10.1 follows directly from the proof of Proposition 10.2.

10.6.2 ∞-norm

Assume instead that we would like to restrict the allowed rotation differently around different axes. For example, if the set of allowed orientations is given by restrictions on the rotation about the x-axis followed by a rotation about the y-axis, this will result in a pyramid-shaped set of allowed directions. The following observations are important in this section.

Rotating the vector $\bar{\boldsymbol{v}}_1 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\mathsf{T}$ by α about the x-axis of the reference frame followed by a rotation β about the y-axis, also of the reference frame, gives the new vector

$$\bar{\boldsymbol{v}}_I = \begin{bmatrix} \cos\alpha \sin\beta \\ -\sin\alpha \\ \cos\alpha\cos\beta \end{bmatrix}. \tag{10.55}$$

For a rotation α about the x-axis of the reference frame followed by a rotation β about the y-axis of the rotated coordinate system, the rotated vector is given by

$$\bar{\boldsymbol{v}}_R = \begin{bmatrix} \sin\beta \\ -\sin\alpha\cos\beta \\ \cos\alpha\sin\beta \end{bmatrix}. \tag{10.56}$$

This can also be written as a quaternion Q. Let the vector $\bar{\boldsymbol{v}}_1$ be rotated by Q into $\boldsymbol{v}_2 = Q * \boldsymbol{v}_1 * Q^*$. Then $\boldsymbol{v}_2$ is written as

$$\boldsymbol{v}_2 = \begin{bmatrix} 0 \\ 2(q_0q_2 + q_1q_3) \\ 2(q_2q_3 - q_0q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{10.57}$$

Proposition 10.3. *Given a restriction α_{lim} in the orientation error about the x-axis of the reference frame and β_{lim} in the orientation error about the y-axis of the rotated coordinate frame. Then the z-axis of $\mathcal{F}_Q$ rotated by the quaternion $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ with respect to the reference frame $\mathcal{F}_I$ lies within the restrictions given by β_{lim} where $\beta_{lim} \geq 0$, if and only if*

$$P_1 = \begin{bmatrix} \eta & 0 & q_1 \\ 0 & \eta & q_0 \\ q_3 & q_2 & \eta \end{bmatrix} \geq 0 \tag{10.58}$$

where $\eta = \sqrt{\frac{\sin\beta_{lim}}{2}}$ and $\geq$ means positive semi-definiteness for the non-symmetric matrix P_1.

Proof. The determinant of P_1 is given by

$$\det(P_1) = \eta(\eta^2 - q_0q_2 - q_1q_3). \tag{10.59}$$

Assume $\det(P_1) \geq 0$

$$\begin{aligned} \eta^2 - q_0q_2 - q_1q_3 &\geq 0 \\ \sin\beta_{lim} &\geq 2(q_0q_2 + q_1q_3). \end{aligned} \tag{10.60}$$

As $\beta_{lim} \geq 0$, comparing Equations (10.56) and (10.57) gives

$$\beta = \arcsin\left(2(q_0 q_2 + q_1 q_3)\right), \tag{10.61}$$

and the initial requirement is obtained by

$$\beta \leq \beta_{lim} \tag{10.62}$$

where β is the angle between the new z-axis and the yz-plane. Similarly for the lower bound. □

Proposition 10.4. *Given a restriction α_{lim} in the orientation error about the x-axis and β_{lim} in the orientation error about the y-axis, both in the reference frame. Then the z-axis of frame $\mathcal{F}_Q$ rotated by the quaternion $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$ with respect to the reference frame $\mathcal{F}_I$ lies within the restrictions given by α_{lim} if*

$$P_2 = \begin{bmatrix} \xi & q_2 q_3 & 0 \\ q_2 q_3 & \xi & q_0 q_1 \\ 2\xi & q_0 q_1 & \xi \end{bmatrix} \geq 0 \tag{10.63}$$

where $\xi = \frac{\sin \alpha_{lim}}{2}$.

Proof. We start with the principal minors and see that we need to add the constraint

$$\xi^2 - q_2^2 q_3^2 > 0. \tag{10.64}$$

The determinant of P_2 is given by

$$\det(P_2) = \xi(\xi^2 - (q_0 q_1)^2 - (q_2 q_3)^2 + 2 q_0 q_1 q_2 q_3). \tag{10.65}$$

Then $\det(P_1) \geq 0$ becomes

$$\begin{gathered} \xi^2 - (q_0 q_1)^2 - (q_2 q_3)^2 + 2 q_0 q_1 q_2 q_3 \geq 0 \\ \xi^2 \geq (q_2 q_3 - q_0 q_1)^2 \\ \sin \alpha_{lim} \geq 2|q_2 q_3 - q_0 q_1|. \end{gathered} \tag{10.66}$$

As $\alpha_{lim} \geq 0$, comparing Equations (10.57) and (10.55) gives

$$\alpha = \arcsin\left(2(q_2 q_3 - q_0 q_1)\right), \tag{10.67}$$

and the initial requirement is obtained by

$$-\alpha_{lim} \leq \alpha \leq \alpha_{lim}. \tag{10.68}$$

□

Note that in Proposition 10.3 the second rotation is with respect to the rotated coordinate frame and the constraints restrict only the rotations about the y-axis while in Proposition 10.4 the second rotation is with respect to the rotated coordinate frame and the

constraints restrict the allowed rotations about the x-axis only. This simplifies the computations substantially and is a good approximation to rotating around the x- and y-axes of $\mathcal{F}_I$. We also note that the matrices given in Propositions 10.3 and 10.4 are not symmetric and that P_2 in (10.63) is not affine. Hence, the constraints cannot be represented as LMIs. They can, however, be represented as barrier functions given as the negative logarithm of the determinant for which we also omit the additional constraint (10.64).

Example 16. Given a restriction α_{lim} in the orientation error about the x-axis and β_{lim} in the orientation error about the y-axis. Then the z-axis of frame $\mathcal{F}_Q$ rotated by the quaternion $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ with respect to the reference frame $\mathcal{F}_Q$ lies within the restrictions given by α_{lim} and β_{lim} if

$$P = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} \geq 0 \tag{10.69}$$

where P_1 and P_2 are given as in Propositions 10.3 and 10.4 respectively.

Alternatively, an accurate solution can be found by restricting the orientation about the x-axis followed by the orientation about the y-axis, also in $\mathcal{F}_I$. This can be achieved by writing $\alpha = \arctan2(\frac{q_2 q_3 + q_0 q_1}{q_0 q_2 - q_1 q_3})$ and substituting

$$\eta = \sqrt{\frac{\cos\alpha \sin\beta_{lim}}{2}} \tag{10.70}$$

for η in Equation (10.58).

10.6.3 Restriction on the Orientation about the Central Axis

We now turn to the pointing task problem, i.e. to determine the rotation about the central axis itself. This will not change the direction of the central axis and thus not influence the orientation error. Assume we want the x-axis to point in one given direction in order to improve performance. This direction may be different at every time step. Also for the x-axis we may allow a small error from the desired direction. For the ZYZ-sequence the direction of the x-axis is given by both α, β and γ. We assume the error of the direction of the z-axis is restricted as in Section 10.6.1. When this is constrained to be relatively small, the error in the direction of the x-axis can be approximated by the error in the orientation about the central axis. This error is given by (10.39) as

$$\epsilon = \alpha + \gamma. \tag{10.71}$$

Proposition 10.5. *Assume that the error in the direction of the z-axis is small. Given a restriction in the orientation error ϵ_{lim} around the central axis, the x-axis of $\mathcal{F}_Q$ rotated by $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ from the reference frame $\mathcal{F}_I$ lies within the restrictions given by $\epsilon_{lim} \geq 0$ if and only if*

$$P = \begin{bmatrix} \kappa & \frac{q_3}{q_0} \\ \frac{q_3}{q_0} & \kappa \end{bmatrix} \succeq 0 \tag{10.72}$$

where $\kappa = \tan\frac{\epsilon_{lim}}{2}$.

Proof. The determinant of P is given by

$$\det(P) = \kappa^2 - \frac{q_3^2}{q_0^2}. \tag{10.73}$$

As ϵ_{lim} is positive, we have $0 \leq \tan\frac{\epsilon_{lim}}{2}$ for $0 \leq \epsilon_{lim} \leq \pi$ and $\det(P) \geq 0$ can be written as

$$\begin{aligned} \kappa^2 &\geq \frac{q_3^2}{q_0^2} \\ \epsilon_{lim} &\geq |2\arctan(\frac{q_3}{q_0})| \end{aligned} \tag{10.74}$$

Then Equation (10.39) concludes that

$$-\epsilon_{lim} \leq \epsilon \leq \epsilon_{lim}. \tag{10.75}$$

□

Also for Proposition 10.5 we can reformulate the result and obtain a barrier function.

Corollary 10.2. Assume that the orientation error of the direction of the z-axis is small and the orientation error about the central axis is restricted to ϵ_{lim} and let $\kappa = \tan(\frac{\epsilon_{lim}}{2})$. Then the barrier function

$$\phi = -\log\left(\kappa^2 - \frac{q_3^2}{q_0^2}\right) \tag{10.76}$$

increases exponentially to infinity as the orientation approaches the orientation limit, forcing the x-axis of $\mathcal{F}_Q$ rotated by $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ from the reference frame $\mathcal{F}_I$ to lie within the restrictions given by ϵ_{lim}.

10.6.4 Direction of the x-axis

Alternatively, one might want to restrict the direction of the x-axis directly. Note that the matrix given in the previous section is not affine and cannot be written as an LMI. Hence, another matrix that is both symmetric and affine is proposed in the following. Assume that the direction of the x-axis is to be restricted. Similarly to Equation (10.49), the requirement that the body frame x-axis is to point in the direction of the reference frame x-axis is given by

$$P_2 = \begin{bmatrix} \xi & 0 & q_2 \\ 0 & \xi & q_3 \\ q_2 & q_3 & \xi \end{bmatrix} \succeq 0 \tag{10.77}$$

where $\xi = \sin\frac{\epsilon_{lim}}{2}$. This will restrict the x-axis of $\mathcal{F}_Q$ to lie within a cone with the x-axis of $\mathcal{F}_I$ at the centre.

This quaternion volume can also be transformed by Equation (10.9). Assume that the direction of the body frame x-axis is to point in the direction given by the direction of the x-axis of $Q_d = \begin{bmatrix} d_0 & d_1 & d_2 & d_3 \end{bmatrix}^\mathsf{T}$. In order to apply the restriction given by (10.77),

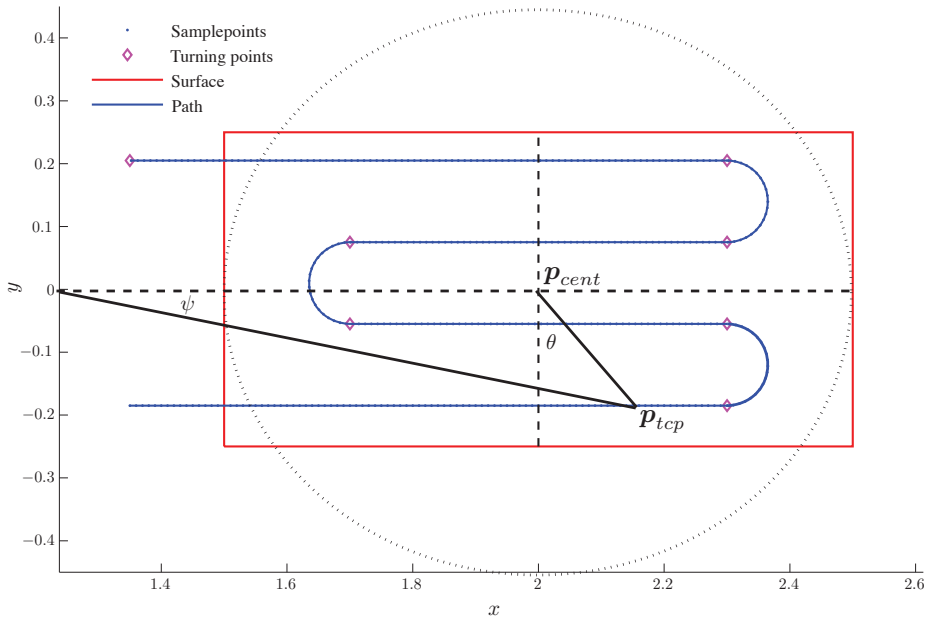


Figure 10.3: The path of the tool centre point (TCP) in the xy-plane. The direction of the central axis is determined from θ by the quaternion $Q_d(\theta)$ and the rotation around the central axis itself is determined from ψ.

but to the direction of the x-axis of $\mathcal{F}_{Q_d}$ and not that of $\mathcal{F}_I$, Q is transformed back into the reference frame and the test is performed on the transformed quaternion

$$Q_t = Q * Q_d^* = \begin{bmatrix} * \\ * \\ -q_0 d_2 + q_2 d_0 - q_3 d_1 + q_1 d_3 \\ -q_0 d_3 + q_3 d_0 - q_1 d_2 + q_2 d_1 \end{bmatrix} \tag{10.78}$$

Note that when Q_t is substituted into (10.77), P_2 is still symmetric and affine in Q.

10.7 Spray Painting

We now show an example where the direction of the z-axis is determined by two cone-shaped sets of orientations. The direction given by the two sets at each time step is in general conflicting and the solution is the minimum of a cost function given by the sum of the two orientation errors. There are two main criteria that will guarantee uniform paint coating, the orientation of the spray gun with respect to the surface and its velocity. The

first restriction is ensured by the constraint

$$\eta^2 - q_1^2 - q_2^2 > 0 \tag{10.79}$$

where $\eta = \sin\left(\frac{\beta_{lim}}{2}\right)$ and β_{lim} is the maximum allowed orientation error for which the quality of the paint job is satisfying. The paint gun should always be orthogonal to the surface, but in general an orientation error of about 20° guarantees uniform paint coating. We will assume a manipulator that is to paint a surface in the xy-plane following the path in Fig. 10.3. The restrictions on the orientation is visualised by a cone. The cross section of this cone is given by the circle in Fig. 10.3.

The second restriction is on the velocity of the paint gun and can be improved by a similar constraint. The general idea is to reduce the displacement of the paint gun by choosing a desired orientation at each time step which forces the position of the paint gun to remain at the centre of the surface. This will reduce the torques in the main axes as these are mainly used for positioning the end effector. Assume we want to paint the surface in the xy-plane with a constant distance z_{des} between the tool and the surface. Let $\boldsymbol{c}$ be the vector from the centre of the surface, at height z_{des}, denoted $\boldsymbol{p}_{cent}$, to the current position $\boldsymbol{p}_{tcp}$ on the surface

$$\boldsymbol{c} = \boldsymbol{p}_{tcp} - \boldsymbol{p}_{cent}. \tag{10.80}$$

This is the direction of the central axis for which the main axes don't need to move at all, i.e. pure rotation of the wrist. We choose this as the desired direction of the central axis when the orientation error is not considered, represented by Q_d. We now introduce the same freedom in this constraint as we did with the orientation error, forcing the orientation to lie inside a quaternion volume with the z-axis of Q_d at the centre.

First we transform the quaternion back into the reference frame and perform the test on the transformed quaternion in the reference frame. The transformed quaternion is given by

$$Q_p(t) = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = Q * Q_d^*(t) = \begin{bmatrix} * \\ -q_0 d_1 + q_1 d_0 - q_2 d_3 + q_3 d_2 \\ -q_0 d_2 + q_2 d_0 - q_3 d_1 + q_1 d_3 \\ * \end{bmatrix}. \tag{10.81}$$

The constraint that forces the end effector to point in the direction of Q_d with a maximum orientation error α_{lim} is given by Proposition 10.2 as

$$\xi^2 - p_1^2 - p_2^2 > 0 \tag{10.82}$$

where $\xi = \sin\left(\frac{\alpha_{lim}}{2}\right)$. Thus, we use the same constraint as for the reference frame, but on the transformed quaternion $Q_p(t)$.

We now turn to the problem of spray painting the surface in the xy-plane in Fig. 10.3, also addressed in From and Gravdahl (2007a). The surface is to be painted from above, so the set representing the orientation error needs to be rotated 180° so that it points downwards. This can be done by to Equation (10.9) with $P = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^\mathsf{T}$ or the approach that we will take here, instead of the restriction $\eta^2 \geq q_1^2 + q_2^2$, which we used in Proposition 10.2, we write

$$\eta^2 \leq q_1^2 + q_2^2, \tag{10.83}$$

and replace $\beta_{lim} \leftarrow \pi - \beta_{lim}$ in $\eta = \sin(\frac{\beta_{lim}}{2})$. This will guarantee that the set of orientations points in exactly the opposite direction of the set of Equation (10.82). The barrier function is then the sum of the two constraints representing the orientation error and the velocity and is given by

$$\begin{aligned}\phi &= k_{err}\phi_{err} + k_{tcp}\phi_{tcp} \\ &= -k_{err}\log(q_1^2 + q_2^2 - \eta^2) - k_{tcp}\log(\xi^2 - p_1^2 - p_2^2).\end{aligned} \tag{10.84}$$

where ϕ_{err} guarantees that the orientation error lies within the limits and ϕ_{tcp} allows the end effector to follow the path with a higher velocity. The weights k_{err} and k_{tcp} weighs the importance of the two restrictions and should be chosen so that the end-effector velocity is constant and as high as possible.

10.7.1 The gradient Method and Implementation

In this section we show how to solve the optimisation problem by the gradient method. The partial derivatives are given by

$$\frac{\partial\phi_{err}}{\partial q_0} = 0, \qquad \frac{\partial\phi_{err}}{\partial q_1} = -\frac{2q_1}{q_1^2 + q_2^2 - \eta^2}, \tag{10.85}$$

$$\frac{\partial\phi_{err}}{\partial q_3} = 0, \qquad \frac{\partial\phi_{err}}{\partial q_2} = -\frac{2q_2}{q_1^2 + q_2^2 - \eta^2}, \tag{10.86}$$

and

$$\frac{\partial\phi_{tcp}}{\partial q_0} = -\frac{2(d_1^2 + d_2^2)q_0 - 2(d_0d_1 + d_2d_3)q_1 + 2(d_1d_3 - d_0d_2)q_2}{\xi^2 - p_1^2 - p_2^2}, \tag{10.87}$$

$$\frac{\partial\phi_{tcp}}{\partial q_1} = -\frac{2(d_0^2 + d_3^2)q_1 - 2(d_0d_1 + d_2d_3)q_0 + 2(d_0d_2 - d_1d_3)q_3}{\xi^2 - p_1^2 - p_2^2}, \tag{10.88}$$

$$\frac{\partial\phi_{tcp}}{\partial q_2} = -\frac{2(d_0^2 + d_3^2)q_2 + 2(d_1d_3 - d_0d_2)q_0 - 2(d_2d_3 + d_0d_1)q_3}{\xi^2 - p_1^2 - p_2^2}, \tag{10.89}$$

$$\frac{\partial\phi_{tcp}}{\partial q_3} = -\frac{2(d_1^2 + d_2^2)q_3 + 2(d_0d_2 - d_1d_3)q_1 - 2(d_2d_3 + d_0d_1)q_2}{\xi^2 - p_1^2 - p_2^2}. \tag{10.90}$$

The gradient is then given by

$$\nabla\phi = \begin{bmatrix} k_{err}\frac{\partial\phi_{err}}{\partial q_0} + k_{tcp}\frac{\partial\phi_{tcp}}{\partial q_0} \\ k_{err}\frac{\partial\phi_{err}}{\partial q_1} + k_{tcp}\frac{\partial\phi_{tcp}}{\partial q_1} \\ k_{err}\frac{\partial\phi_{err}}{\partial q_2} + k_{tcp}\frac{\partial\phi_{tcp}}{\partial q_2} \\ k_{err}\frac{\partial\phi_{err}}{\partial q_3} + k_{tcp}\frac{\partial\phi_{tcp}}{\partial q_3} \end{bmatrix}. \tag{10.91}$$

The problem is solved by the gradient method

$$\phi^{k+1} = \phi^k - a\nabla\phi. \tag{10.92}$$

For a feasible initial condition and for a relatively small and constant step size a the stability and convergence of the method is good. Due to the low computational burden of this approach, a constant step is used instead of a search. This requires that a is chosen conservatively which may lead to slower convergence.

10.7.2 The Pointing Task

By the approach described in the previous section, the orientation about the central axis (z-axis) is not determined. In this section we show how to utilise the last degree of freedom to improve performance further. We will present three different approaches for implementing the solution to the pointing task problem. The orientations found do not differ very much, but the implementations are quite different.

From and Gravdahl (2007a)

The first approach presented is the intuitive approach given in From and Gravdahl (2007a). The orientation about the central axis at point i is set as

$$\psi(t) = k_\psi \arctan2\left(\frac{y(t)}{x(t) - x_{cent}}\right) \tag{10.93}$$

for $k_\psi \in (0, 1]$ and where $x(t)$ and $y(t)$ give the position of the end effector at time t in the xy-plane and x_{cent} is the centre of the surface in the x-direction. $\psi(t)$ is shown in Fig. 10.3. It was shown in From and Gravdahl (2007a) that will reduce the displacement of the main axes.

Direction of the x-axis (Section 10.6.4)

A similar approach is to force the end effector x-axis to point in the direction of the base of the manipulator. By projecting the end-effector x-axis into the xy-plane and force this to point in the direction of the base will have approximately the same effect as the approach in the previous section, but this constraint can easily be written on the form of (10.78) as

$$Q_r(t) = \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = Q * Q_e^*(t) = \begin{bmatrix} * \\ * \\ -q_0e_2 + q_2e_0 - q_3e_1 + q_1e_3 \\ -q_0e_3 + q_3e_0 - q_1e_2 + q_2e_1 \end{bmatrix}, \tag{10.94}$$

where $Q_e(t)$ is time varying and takes the end-effector x-axis into the desired direction. Further, we want the end-effector x-axis to point in the opposite direction of the global x-axis, so we let $\gamma_{lim} \leftarrow \pi - \gamma_{lim}$ and write the corresponding cost function as

$$\phi_x = -\log(r_2^2 + r_3^2 - \nu^2), \tag{10.95}$$

where $\nu = \sin(\frac{\gamma_{lim}}{2})$ and γ_{lim} is the maximum error allowed in the direction of the x-axis. The partial derivatives are given by

$$\frac{\partial \phi_x}{\partial q_0} = -\frac{2(e_2^2+e_3^2)q_0 - 2(e_0e_2+e_1e_3)q_2 + 2(e_1e_2-e_0e_3)q_3}{r_2^2+r_3^2-\nu^2}, \tag{10.96}$$

$$\frac{\partial \phi_x}{\partial q_1} = -\frac{2(e_2^2+e_3^2)q_1 + 2(e_0e_3-e_1e_2)q_2 - 2(e_1e_3+e_0e_2)q_3}{r_2^2+r_3^2-\nu^2}, \tag{10.97}$$

$$\frac{\partial \phi_x}{\partial q_2} = -\frac{2(e_0^2+e_1^2)q_2 - 2(e_0e_2+e_1e_3)q_0 + 2(e_0e_3-e_1e_2)q_1}{r_2^2+r_3^2-\nu^2}, \tag{10.98}$$

$$\frac{\partial \phi_x}{\partial q_3} = -\frac{2(e_0^2+e_1^2)q_3 + 2(e_1e_2-e_0e_3)q_0 - 2(e_1e_3+e_0e_2)q_1}{r_2^2+r_3^2-\nu^2}. \tag{10.99}$$

Thus, the search direction for every time step is given by

$$\nabla\phi = \begin{bmatrix} k_{err}\frac{\partial \phi_{err}}{\partial q_0} + k_{tcp}\frac{\partial \phi_{tcp}}{\partial q_0} + k_x\frac{\partial \phi_x}{\partial q_0} \\ k_{err}\frac{\partial \phi_{err}}{\partial q_1} + k_{tcp}\frac{\partial \phi_{tcp}}{\partial q_1} + k_x\frac{\partial \phi_x}{\partial q_1} \\ k_{err}\frac{\partial \phi_{err}}{\partial q_2} + k_{tcp}\frac{\partial \phi_{tcp}}{\partial q_2} + k_x\frac{\partial \phi_x}{\partial q_2} \\ k_{err}\frac{\partial \phi_{err}}{\partial q_3} + k_{tcp}\frac{\partial \phi_{tcp}}{\partial q_3} + k_x\frac{\partial \phi_x}{\partial q_3} \end{bmatrix}. \tag{10.100}$$

Applying the gradient method will find the minimum of a cost function given by the sum of three in general conflicting objectives. ϕ_{err} guarantees that the orientation error is within its limits, ϕ_{tcp} increases the velocity of the paint gun and ϕ_x exploits the pointing task to increase the velocity further.

Restrictions of the Rotation about the central axis (Section 10.6.3)

By Proposition 10.5, we get that the rotation about the z-axis can be forced to zero by the cost function

$$\phi_x = -k_x \log(\kappa^2 - \frac{q_3^2}{q_0^2}). \tag{10.101}$$

The partial derivatives are given by

$$\frac{\partial \phi_x}{\partial q_1} = 0, \qquad \frac{\partial \phi_x}{\partial q_0} = -\frac{2q_3^2}{q_0(\kappa^2 q_0^2 - q_3^2)},$$
$$\frac{\partial \phi_x}{\partial q_2} = 0, \qquad \frac{\partial \phi_x}{\partial q_3} = \frac{2q_3}{\kappa^2 q_0^2 - q_3^2}.$$

We would like the x-axis to point in the direction of the base, which we obtain by a rotation about the z-axis by $Q_e = \begin{bmatrix} e_0 & 0 & 0 & e_3 \end{bmatrix}^\mathsf{T}$. Again we use $Q_r = Q * Q_e^*$ and

$$\phi_x = -\log(\kappa^2 - \frac{r_3^2}{r_0^2}) \tag{10.102}$$

where

$$Q_r(t) = \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = Q * Q_e^*(t) = \begin{bmatrix} q_0e_0 + q_3e_3 \\ * \\ * \\ -q_0e_3 + q_3e_0 \end{bmatrix}. \tag{10.103}$$

The partial derivatives are then given by

$$\begin{aligned}
\frac{\partial \phi_x}{\partial q_1} &= 0, & \frac{\partial \phi_x}{\partial q_2} &= 0, \\
\frac{\partial \phi_x}{\partial q_0} &= \frac{2(e_3^4 - e_0^4)q_0 q_3^2 - 2(q_3^3 - q_0^2 q_3)(e_0^3 e_3 + e_0 e_3^3)}{r_0^2(\kappa^2 r_0^2 - r_3^2)}, \\
\frac{\partial \phi_x}{\partial q_3} &= \frac{2(e_0^4 - e_3^4)q_0^2 q_3 - 2(q_0^3 - q_0 q_3^2)(e_0 e_3^3 + e_0^3 e_3)}{r_0^2(\kappa^2 r_0^2 - r_3^2)}.
\end{aligned} \tag{10.104}$$

Then by choosing Q_e such that the x-axis points in the direction of the base by a rotation about the z-axis, we obtain the desired motion characteristics. Note that in (10.102) the central axis is assumed to be orthogonal to the surface. Hence, the results are only valid when a small orientation error in the direction of the z-axis is allowed.

10.7.3 LMIs

We now turn to the problem of how to formulate the constraints on the orientation as LMIs and how to solve this when several constraints are present. The problem

$$\begin{array}{ll} \text{minimise} & \phi(x) = \log\det G(x)^{-1} \\ \text{subject to} & G(x) \succ 0 \end{array} \tag{10.105}$$

where

$$G(x) = G_0 + x_1 G_1 + x_2 G_2 + \cdots + x_m G_m, \tag{10.106}$$

is known as the analytic centering problem. This formulation allows us to formulate the restrictions on the z- and x-axes in one big block diagonal matrix and solve this very efficiently. If the feasible set $\boldsymbol{X} = \{x \mid G(x) \succ 0\}$ is non-empty and bounded, the matrices $G_i, i = 1, ..., m$ are linearly independent and the objective function is strictly convex on $\boldsymbol{X}$ (Vandenberghe et al., 1996). In this case, it can be guaranteed that the optimality condition $\nabla\phi(x^*) = 0$, for an optimal solution x^*, can be reached.

In our case, the constraints on the z-axis are written as:

$$\begin{array}{ll} \text{minimise} & \phi(x) = \log\det P(x)^{-1} \\ \text{subject to} & P(x) \succ 0 \end{array} \tag{10.107}$$

where P is given by Equation (10.49) and can be written as

$$P(x) = P_0 + x_1 P_1 + x_2 P_2 + x_3 P_3 + x_4 P_4, \tag{10.108}$$

where

$$x_1 = q_0, \quad x_2 = q_1, \quad x_3 = q_2, \quad x_4 = q_3, \tag{10.109}$$

$$P_0 = \begin{bmatrix} \eta & 0 & 0 \\ 0 & \eta & 0 \\ 0 & 0 & \eta \end{bmatrix}, \qquad P_1 = P_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$P_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \qquad P_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

q_0 and q_3 do not affect the solution and can be eliminated from the equations.

To apply the time varying constraints on the transformed x-axis, substitute (10.78) into (10.77), denote the resulting matrix F, and write it on the form of (10.108) so that

$$F(x) = F_0 + x_1F_1 + x_2F_2 + x_3F_3 + x_4F_4, \tag{10.110}$$

$$x_1 = q_0, \quad x_2 = q_1, \quad x_3 = q_2, \quad x_4 = q_3, \tag{10.111}$$

$$F_0 = \begin{bmatrix} \xi & 0 & 0 \\ 0 & \xi & 0 \\ 0 & 0 & \xi \end{bmatrix}, \qquad F_1(t) = \begin{bmatrix} 0 & 0 & -d_2 \\ 0 & 0 & -d_3 \\ -d_2 & -d_3 & 0 \end{bmatrix},$$

$$F_2(t) = \begin{bmatrix} 0 & 0 & -d_3 \\ 0 & 0 & d_2 \\ -d_3 & d_2 & 0 \end{bmatrix}, \qquad F_3(t) = \begin{bmatrix} 0 & 0 & d_0 \\ 0 & 0 & -d_1 \\ d_0 & -d_1 & 0 \end{bmatrix},$$

$$F_4(t) = \begin{bmatrix} 0 & 0 & d_1 \\ 0 & 0 & d_0 \\ d_1 & d_0 & 0 \end{bmatrix}.$$

To combine the restrictions of the x- and z-axes we use Lemma 10.2 and formulate the problem as

$$\begin{aligned} &\text{minimise} && \phi(x) = \log\det \begin{bmatrix} P(x) & 0 \\ 0 & F(x) \end{bmatrix}^{-1} \\ &\text{subject to} && \begin{bmatrix} P(x) & 0 \\ 0 & F(x) \end{bmatrix} \succ 0 \end{aligned} \tag{10.112}$$

for which the solution is the orientation which minimises the error both of the x-axis and the z-axis with a "metric" that increases exponentially with the angular distance from the desired directions of the x- and z-axes. Also note that for two conflicting constraints on the direction of the z-axis, the constraints given by Equation (10.82) can be written similarly by substituting (10.81) into (10.49).

10.7.4 Normalisation

The optimisation algorithms described optimise freely over all quaternions, and it is thus not guaranteed, nor likely, that the resulting quaternion is of unit length. One simple and very effective, though not very mathematically sound solution, is to optimise freely over all quaternions and then normalise the result afterwards. This turns out to work very well in practice. Another option is to add the constraint $|Q| = 1$ in the optimisation algorithm which guarantees that the search space is only the set of quaternions of unit length.

10.7.5 Optimality and Existence of the Solutions

We note that the quaternion volumes must be chosen so that a solution exists. The quaternion volume representing the orientation error should be chosen according to the maximum allowed error. For the quaternion volume constructed to increase velocity we have more freedom in choosing the size of the volume. This should thus be chosen big enough

so that a solution always exists. This can then be compensated for by increasing k_{tcp} in Equations (10.84) or (10.91).

The optimal orientation at every time step can be found in real time given the velocity of the paint gun. However, the optimal velocity is not found in real time. This is achieved by increasing the velocity until the simulations show that the joint torques reach the limits. Thus, to find the optimal velocity, we need to perform several simulations or test runs to find this. In this sense the solution is not found in real time. On the other hand, if the manipulator is to follow a trajectory for which the maximum velocity is not found by test runs, we can use information about the curvature of the path and the maximum orientation error to choose a velocity that is far higher than for the conventional approach. In this sense, the solution is optimal for the chosen velocity.

The main strength of this method lies in its simplicity. The low computation time allows us to run the problem several times to find a solution very close to the optimal one. There are many alternative approaches well suited to find an optimal or closer to optimal solution. A learning approach may find a more optimal solution, but this would require far more computational effort. One might also construct an optimisation problem that optimises the torques given a freedom in the orientation, but to find an optimal global solution to this problem is extremely complex. The short computation time for the proposed algorithm makes it a good alternative to the computationally more demanding approaches.

10.7.6 Curved Surfaces

The approach presented is not limited to planar surfaces. For curved surfaces such as the hood of a car, we can use the exact same approach. The desired direction of the end effector used in (10.81) can be chosen as the same as the planar case. However, the quaternion volume representing the orientation error must be transformed similarly to (10.81) so that the centre of the volume is orthogonal to the surface at every point on the trajectory. For curved surfaces we expect the performance to improve more than for a plane as the orientation of the paint gun does not have to follow the optimal orientation (orthogonal to the surface) as tightly and can sweep over the surface more smoothly and with less variation in the orientation.

10.8 Numerical Examples

10.8.1 Convergence

Table 10.1 shows the computational efficiency of the algorithms presented. The convergence is in general very good and a solution is found in 10-20 iterations. In some cases a few more iterations are needed, but for all the tests performed, about 50 iterations is sufficient, as a worst-case measure. No information from the previous solution is used in choosing the initial conditions. The simulations were performed on an Intel T7200 2GHz processor. We can see that the time needed for each iteration is very low. Even for the worst case of 50 iterations the time needed to find a solution is less than one millisecond. This makes all the algorithms presented suitable for on-line implementation.

The three algorithms presented were compared in terms of computational efficiency. The algorithms tested were i) z-axis cone restrictions as presented in Section 10.7.1; ii)

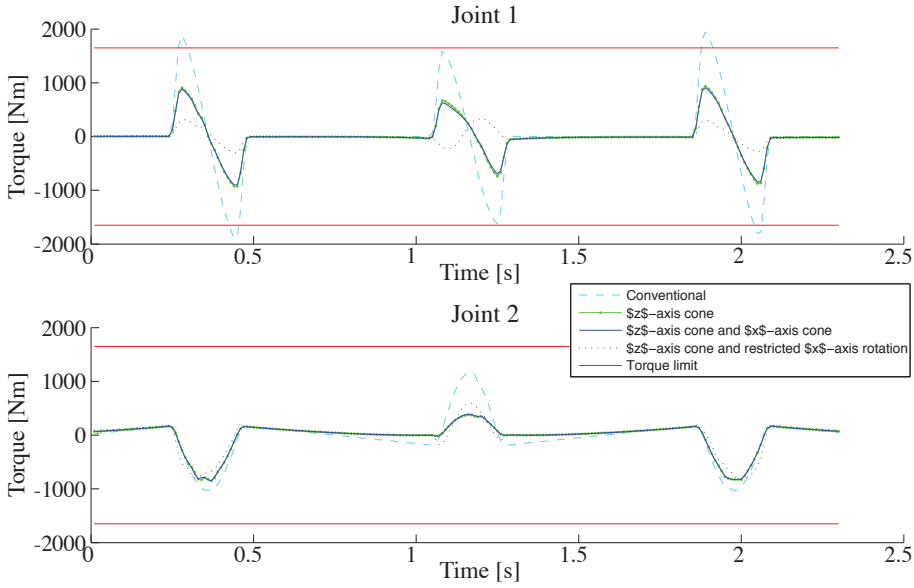


Figure 10.4: Torques for joint 1 and 2 for the four different approaches presented.

z-axis cone restrictions as presented in Section 10.7.1 with additional cone restriction on the direction of the x-axis as presented in Section 10.7.2; and iii) z-axis cone restrictions as presented in Section 10.7.1 with additional restriction on the rotation about the z-axis as presented in Section 10.7.2.

Algoritm	Iteration time [ms]	Max its needed	Max time [ms]
i) z-axis cone	0.00232	50	0.116
ii) z-axis cone & x-axis cone	0.00268	50	0.1608
iii) z-axis cone & restr x-axis	0.00605	50	0.363

Table 10.1: Speed for one iteration, number of iterations needed to "guarantee" an optimal solution (worst case), and time needed to obtain optimal solution.

10.8.2 Trajectory Speed

The same algorithms were tested for trajectory following. The manipulator was to follow the path given in Fig. 10.3 with a constant speed of 1m/s. The torques of joints 1 and 2 for each case is shown in Fig. 10.4 together with the torque limits of each joint. We can see that all the approaches improve performance substantially. The approach that only adds constraints on the direction of the z-axis performs very well and is very easy to implement. For large allowed orientation errors of the z-axis the x-axis cone will reduce

the orientation error not only of the x-axis but also the z-axis. This may be considered a side-effect of this cone constraint as the main motivation behind this restriction is to change the direction of the x-axis and not the z-axis. This side-effect is not present for the last approach which determines the direction of the x-axis by restricting the rotation around the end-effector z-axis. This approach will thus perform better in some cases as the orientation error of the z-axis, which is our main concern, is not reduced. This approach does, however, have a numerical singularity when q_0 approaches zero. This must be handled in the implementation.

Table 10.2 shows the maximum speed for which the manipulator can follow the path for each algorithm. The speed increases for all the approaches presented. Table 10.2 also shows the maximum orientation error of the z-axis in each case. The maximum allowed orientation error is set to 20° for all approaches. We see that the maximum orientation error when both the z- and x-axes are restricted by a cone is lower than for the two other cases. This is because, as described above, the restriction on the x-axis cone will also affect direction of the z-axis. As the direction of the z-axis is our main tool to improve performance, this approach does not perform as well as the other two when large orientation errors are allowed.

Algorithm	Max vel [m/s]	Max or. error [°]
Conventional	0.91	0
i) z-axis cone	1.35	20
ii) z-axis cone & x-axis cone	1.28	12
iii) z-axis cone & restr x-axis	1.37	20

Table 10.2: The maximum speed the manipulator can follow the path for the four different approaches and the corresponding orientation errors.

10.9 Conclusion

In this paper we have shown how to transform a constraint on a continuous set of orientations into a convex constraint. By representing the constraints as LMIs or barrier functions the optimal solution for a given cost function can be found in real time at every time step. For spray paint applications this allows us to exploit the fact that a small orientation error can be utilised to increase the velocity of the paint gun during turn, guaranteeing uniform paint coating and substantially decreasing the time needed to paint a surface.

10.10 Appendix

The quaternion volume is given by Equation (10.22), i.e.

$$\begin{bmatrix} \cos(\frac{\beta}{2})\cos(\frac{\alpha}{2}+\frac{\gamma}{2}) \\ \sin(\frac{\beta}{2})\sin(\frac{\alpha}{2}-\frac{\gamma}{2}) \\ \sin(\frac{\beta}{2})\cos(\frac{\alpha}{2}-\frac{\gamma}{2}) \\ \cos(\frac{\beta}{2})\sin(\frac{\alpha}{2}+\frac{\gamma}{2}) \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \begin{matrix} (I) \\ (II) \\ (III) \\ (IV) \end{matrix} \tag{10.113}$$

By substituting (II) into (III), (III) becomes

$$\sin(\frac{\beta}{2})\sqrt{1-\frac{q_1^2}{\sin^2(\frac{\beta}{2})}}=q_2 \tag{10.114}$$

and $\beta = 2\arcsin\sqrt{q_1^2+q_2^2}$ is positive by definition. α and γ are found by dividing (II) by (III) and (IV) by (I):

$$\tan(\frac{\alpha}{2}-\frac{\gamma}{2})=\frac{q_1}{q_2}, \qquad \tan(\frac{\gamma}{2}+\frac{\alpha}{2})=\frac{q_3}{q_0}. \tag{10.115}$$

We write

$$\frac{\alpha}{2}-\frac{\gamma}{2}=\arctan(\frac{q_1}{q_2}), \qquad \frac{\gamma}{2}+\frac{\alpha}{2}=\arctan(\frac{q_3}{q_0}), \tag{10.116}$$

so that α, β and γ are given by Equations (10.36)-(10.38).

Chapter 11

Optimal Paint Gun Orientation in Spray Paint Applications - Experimental Results

11.1 Abstract

In this paper we present the experimental results of a new spray paint algorithm presented in previous publications. Both theory and simulations indicate that the proposed method allows a robotic manipulator to paint a given surface using substantially lower joint torques than with conventional approaches. In this paper we confirm this by implementing the algorithm on an ABB robot and we find that the joint torques needed to follow the trajectory are substantially lower than for the conventional approach.

The approach presented is based on the observation that a small error in the orientation of the end effector does not affect the quality of the paint job. It is far more important to maintain constant velocity for the entire trajectory. We thus propose to allow a small error in the specification of the end-effector orientation and show how this allows us to obtain a higher constant speed throughout the trajectory. In addition to improve the uniformity of the paint coating we are able perform the paint job in less time.

***Note to Practitioners*—This paper is motivated by the observation that uniform paint coating cannot be achieved in steep turns. Even if the manipulator possesses the necessary actuator torques to maintain constant speed for a straight line trajectory the torques needed to maintain constant velocity during turn are far higher. Thus, the operator has to lower the trajectory speed, also in the straight line segments where this would normally not be necessary, or accept a thicker layer of paint in the turns. The method proposed in this paper is to implement a slightly different planning algorithm in turns allowing the paint gun to follow the trajectory with a higher constant velocity.**

Keywords—Spray painting, assembly-line manufacturing, modelling, empirical studies.

11.2 Introduction

In robotics research empirical studies are extremely important in order to validate algorithms and simulation results. Even though simulation tools are becoming increasingly accurate, they can never compare to real-world experiments. In this paper we present the empirical data obtained by implementing three different spray paint algorithms and running these on a robot manipulator in our lab. The algorithms compute a trajectory in joint space for which the end-effector follows a pre-defined path. The joint torques are then measured for the different approaches and compared to the conventional approach. The details of the proposed approach and the expressions to be implemented are found in From and Gravdahl (2010b).

In Potkonjak et al. (2000) the idea of introducing the paint quality as a constraint and minimise some additional cost function was presented. This opens for the possibility of allowing a small error in the orientation of the end effector in order to increase the velocity of the paint gun, reduce torques, and so on. In Buss et al. (1996) the problem of friction force limit constraints was transformed into a problem of testing for positive definiteness of a certain matrix and in From and Gravdahl (2008d) the same ideas were used to convert the problem of orientation error constraints into a test of positive definiteness of a matrix. For different types of orientation errors, a suitable matrix was found and it was shown that positive definiteness of this matrix is equivalent to an orientation satisfying the given restrictions. By transforming the non-linear orientation constraints into positive definiteness constraints imposed on certain matrices we transformed the problem of finding the optimal orientation into an optimisation problem on the smooth manifold of linearly constrained positive definite matrices (From and Gravdahl, 2010b).

In From and Gravdahl (2010b) we showed that by allowing an orientation error of about 20° we are able to 1) reduce the torques required to follow a path by about 50% and 2) increase the speed at which the end effector can follow the trajectory with about 50%. These results were found through simulations. It is, however, important to confirm these results also through empirical studies in the lab. In this paper we have implemented the same algorithms on a robot and measured the torques needed to follow a typical path for painting a flat and a curved surface.

11.3 Problem Statement

There are two main factors that play important roles in obtaining uniform paint coating in automotive manufacturing. The first is to move the paint gun with constant velocity throughout the trajectory. This is in general an easy task in following straight lines but can be a challenge in turns where high accelerations are required. The second factor is the orientation of the paint gun with respect to the surface, which should be orthogonal.

It can be shown that the velocity of the paint gun is far more important than the orientation when it comes to uniform paint coating. A small error ($<20^\circ$) in the paint gun orientation does not affect the quality of the coating to the same extent as changes in the velocity. Based on these observations we represent the orientation not as one frame, but as a constrained continuous set of frames. The problem treated in this paper is then formulated as follows:

Given a maximum allowed orientation error of the paint gun and a trajectory on the surface that the paint gun is to follow with constant velocity and with a fixed distance from the paint gun to the surface. Then the problem is to find the orientation of the paint gun at every point on the trajectory that allows it to follow the trajectory with the highest possible constant velocity.

We consider a standard industrial manipulator. The first three joints are referred to as the main axes, or the main joints. These are the strongest joints and also the ones that require the most torque. While the main axes are mainly used for positioning the paint gun, the last three joints, referred to as the wrist joints, determine the orientation of the paint gun. We fix the inertial reference frame to the base of the manipulator. We also attach a frame to the end effector of the manipulator, in our case the paint gun. This is attached so that the end-effector z-axis is aligned with the direction of the paint flow. This axis is referred to as the central axis.

We thus follow the standard approach for defining the reference frames of the manipulator but instead of specifying only one frame, we can define a continuous set of frames that lie close to the original frame by some metric. The algorithm is then free to choose any frame that lies sufficiently close to the original frame. At each point in the path, the proposed algorithm then chooses the frame that results in the highest possible constant speed of the end effector.

For a spray paint robot following a path as the one in Fig. 11.1 the main work load is on the main axes of the robot, i.e., joints 1, 2, and 3. We observe that the work load on the wrist axes is very small compared to the main axes. The proposed algorithm will thus endeavour to move some of the work load from the main axes to the wrist axes. Because the joint torques of the main axes are very close to the torque limits, this should allow us to follow the trajectory with a higher velocity. Increasing the torques of the wrist axes should not pose any problems as these are very small for this type of trajectory.

11.4 Experimental Set-up

All the experiments were performed in ABB's robot lab in Oslo, Norway on the ABB IRB-4400 industrial robot. The robot was equipped with an end effector of approximately the same weight as a spray paint gun. The optimal trajectories were computed off-line to allow for analysis, but as far as the computation time is concerned, the computations could have been performed on-line. The resulting optimal trajectories were implemented in joint space feeding the joint positions for each joint at constant time intervals.

During the experiments the positions, velocities, and torques of each joint were saved. All signals are scaled so that the maximum value equals 1. The measurements for the three different approaches are scaled by the same factor, so the plots presented in the next section are comparable and illustrate well the difference between the different approaches.

The robot was set to follow a path as the one shown in Fig. 11.1 which consists of four straight line segments and three turns. p_{tcp} denotes a position at the surface and at every time t the spray gun must point in the direction of $p_{tcp}(t)$. This is the path of the tool centre point (TCP). We perform the experiments for two different surfaces: 1) a flat surface in the xy-plane, and 2) a curved surface. The surfaces are illustrated in Fig. 11.1. All the experiments are performed with a constant TCP velocity of 0.9 m/s.

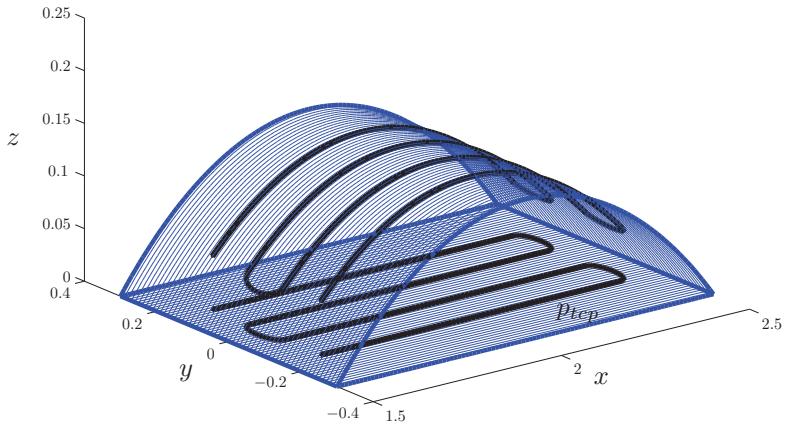


Figure 11.1: The path of the tool centre point (TCP) for the flat surface in the xy-plane and the curved surface.

11.5 Experimental Results

In this section we present the experimental results when the trajectory is computed using the following algorithms:

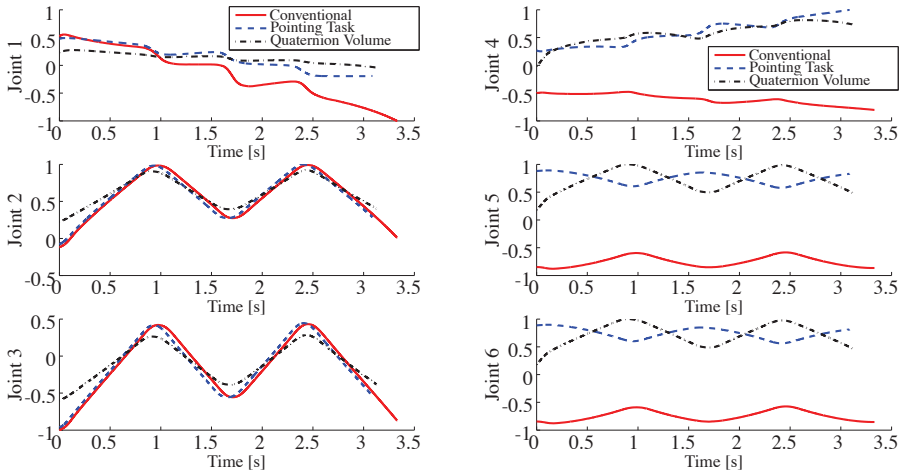
1. **Conventional** - the orientation error is zero and the optimal orientation around the central axis is not utilized to improve performance.

2. **Pointing Task** - the orientation error is zero and the optimal orientation around the central axis is found. The rotation about the central axis can be chosen freely.

3. **Quaternion Volume** - an orientation error of 20° is allowed and the optimal orientation is found. We also optimise around the central axis, as in 2).

11.5.1 Flat Surface

The joint positions for the main and wrist axes are found in Fig. 11.2. We can see that the joint trajectories found by the three algorithms are quite different even though the end-effector position is basically the same and the orientation differs only slightly for the different approaches. The trajectories in joint space are quite similar for joint 2 and 3, which is due to the kinematic coupling between these joints in the IRB-4400. The corresponding joint velocities are found in Fig. 11.3. Fig. 11.4 shows the power for all the joints. We see clearly that the energy used is reduced for the main axes and that the wrist axes take more of the work load. The largest reduction is found in the first joint, which



(a) The main axes.

(b) The wrist axes.

Figure 11.2: Joint positions. All positions are scaled.

corresponds well with the position and velocity plots. We also note that for the wrist axes we use considerably more energy when an orientation error is allowed.

	Conventional	Pointing Task	Quat Volume
1	0.363	0.162	0.066
2	1.000	0.881	0.518
3	0.416	0.410	0.212
4	0.010	0.012	0.021
5	0.017	0.053	0.032
6	0.011	0.014	0.011

Table 11.1: The square of the torques for each joint. All values are scaled.

We use the square of the torque over the trajectory as a metric to compare the amount of torque needed to follow the three paths. This is given for each joint in Table 11.1. We see that the torques needed to follow the trajectory for the main axes decrease for the pointing task and even further for the quaternion volume. To show this more clearly, Table 11.2 shows the average torques for the main and wrist axes. We see that for the main axes the torques needed to follow the trajectory are reduced by 18% for the pointing task and 55% for the quaternion volume. This shows that the proposed algorithms work well and, as anticipated, the work load on the main axes are reduced substantially. For the wrist axes, however, the torques increase. This is as expected, since the main objective was to move the work load from the main axes to the wrist axes. However, as the wrist torques needed to follow the trajectory for the conventional approach were so small, these are

still well inside the torque limits and can still be considered very small, even though they increase by 117% and 83% for the pointing task and the quaternion volume, respectively. The average joint torque is shown in Table 11.3.

	Conventional	Pointing Task	Quat Volume
Main	0.593	0.484 (-18.4%)	0.265 (-55.3%)
Wrist	0.012	0.026 (+116.6%)	0.022 (+83.3%)

Table 11.2: The average of the square of the torques for the main and wrist axes.

	Conventional	Pointing Task	Quat Volume
All Joints	0.303	0.255	0.143

Table 11.3: The average of the square of the torques for all the joints.

The maximum and minimum torques are shown in Table 11.4. We see that for the main axes also the maximum values decrease for the pointing task and quaternion volume. As for the average torques, the maximum torques increase for the wrist axes. Again, these results are as expected and the work load is moved from the main axes to the wrist axes.

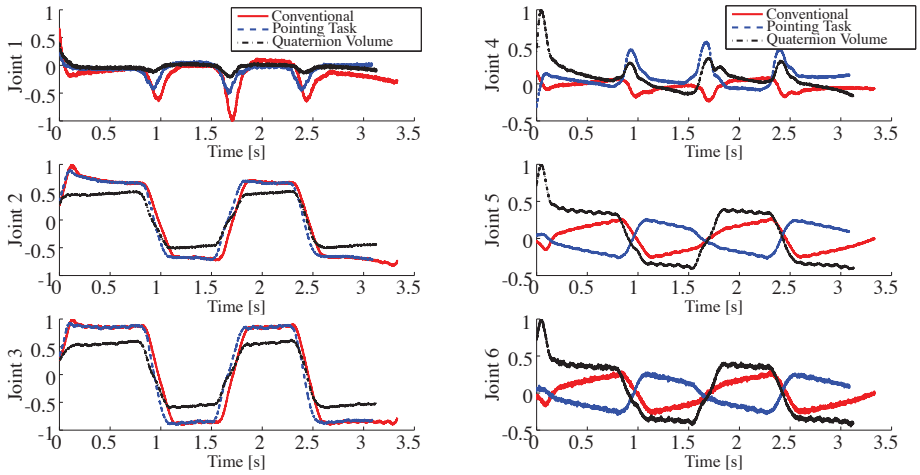
	Conventional		Pointing Task		Quat Volume	
Joint	Max	Min	Max	Min	Max	Min
1	1.000	-0.483	0.752	-0.355	0.555	-0.211
2	0.791	-0.553	0.789	-0.667	0.700	-0.647
3	0.452	-0.486	0.520	-0.592	0.329	-0.495
4	0.087	-0.066	0.174	-0.053	0.176	-0.290
5	0.065	-0.069	0.433	-0.100	0.191	-0.160
6	0.043	-0.069	0.158	-0.047	0.075	-0.073

Table 11.4: The average of the maximum and minimum joint torques for the different approaches. All values are normalised.

	Conventional		Pointing Task		Quat Volume	
	Max	Min	Max	Min	Max	Min
Main	0.745	-0.507	0.682	-0.540	0.528	-0.418
Wrist	0.065	-0.061	0.255	-0.067	0.147	-0.175

Table 11.5: The average of the maximum and minimum of the main and wrist axes.

The maximum velocities for which we can follow the trajectory in Fig. 11.1 are shown in Table 11.7. We see that by optimising the orientation around the central axis (the point-



(a) The main axes.

(b) The wrist axes.

Figure 11.3: Joint velocities. All velocities are scaled.

ing task) at every time step, we can increase the maximum speed slightly without allowing an orientation error. Any planning algorithm should thus include an optimisation around the central axis to be able to increase the maximum speed or simply reduce the joint torques. However, if we allow an orientation error of about 20° we see that we can increase the maximum speed with 50% compared to the convetional approach. Allthough the numbers are slightly different from the simulation results, the ratio between the numbers are approximately the same and the experimental results thus confirm the simulations presented in From and Gravdahl (2010b).

	Conventional		Pointing Task		Quat Volume	
	Max	Min	Max	Min	Max	Min
All Joints	0.406	-0.284	0.471	-0.302	1.337	-0.296

Table 11.6: The average of the maximum and minimum torques.

11.5.2 Curved Surface

The approach presented is not limited to planar surfaces. For curved surfaces such as the hood of a car, we can use the same approach. The path of the tool centre point is the same and the direction of the end effector, assuming no orientation error, is set orthogonal to the surface at each point in the TCP-path.

We see from Tables 11.8 and 11.9 that for the curved surfaces we can reduce the torques needed to follow the path even more than for flat surfaces. The torques needed to follow

	Max vel [m/s]	Max or. error [°]
Conventional	0.94	0
Pointing Task	0.98	0
Quaternion Volume	1.41	20

Table 11.7: The maximum speed the manipulator can follow the path and the corresponding orientation errors.

	Conventional	Pointing Task	Quat Volume
Main	0.623	0.509 (-18.3%)	0.271 (-56.5%)
Wrist	0.152	0.232 (+52.6%)	0.202 (+32.9%)

Table 11.8: The average of the square of the torques for the main joints and the wrist joints when applied to a curved surface.

	Conventional		Pointing Task		Quat Volume	
	Max	Min	Max	Min	Max	Min
Main	0.904	-0.677	0.792	-0.702	0.596	-0.520
Wrist	0.277	-0.223	0.288	-0.167	0.224	-0.201

Table 11.9: The average of the maximum and minimum of the main and wrist axes when applied to a curved surface.

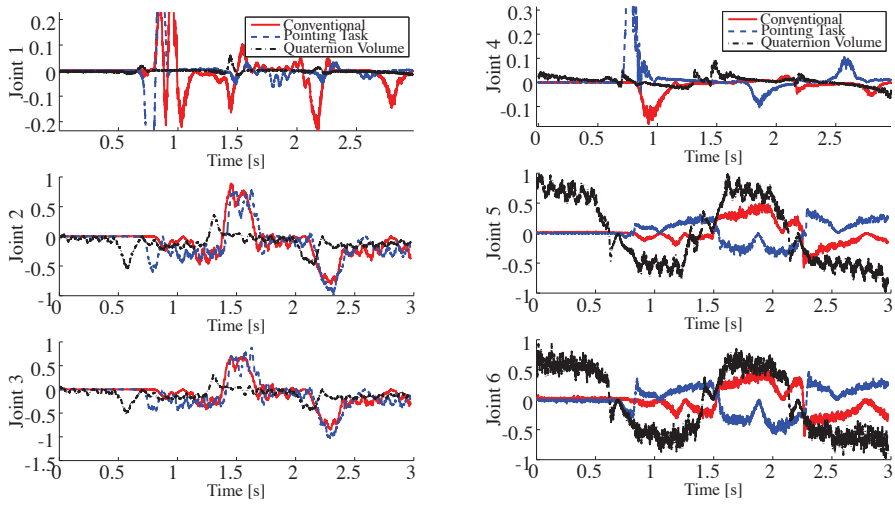
a curved path are somewhat larger than for flat surfaces using the conventional approach, but at the same time the gain that we get from allowing an orientation error is larger and the torques needed to follow the trajectory on a flat and curved surface are more or less the same when an orientation error of 20° is allowed.

As expected the performance of the algorithm improves for a curved surface compared to the plane. This is mainly because for the curved surface the orientation of the paint gun changes over the path. Allowing a freedom in the specifications of the end-effector orientation allows us to "even out" these changes in the orientation and thus sweep over the surface more smoothly.

11.6 Conclusion

The algorithm in this paper is based on the observation that a small error in the end-effector orientation does not decrease the quality of the paint job to a large extent. To guarantee uniform paint coating, it is far more important to maintain constant velocity throughout the trajectory. We thus propose to use the freedom that arises when we allow a small orientation error to increase the velocity of the end effector.

Previous publications have shown that one should be able to reduce the maximum torques and the energy needed to follow a specific path by about 50% by allowing a small



(a) The main axes.

(b) The wrist axes.

Figure 11.4: The power of each joint. All plots are scaled.

orientation error in the specification of the end effector. The need to confirm these promising simulation results through experiments is thus apparent. In this paper we have validated the theory and simulations presented previously and shown that we can substantially reduce the joint torques needed for a spray paint robot to follow a specific end-effector trajectory. We have shown that both the energy used and the maximum torques are reduced. This allows us to paint the surface considerably quicker than with the conventional approach.

In this paper we have also investigated how the algorithm performs on curved surfaces. As expected we are able to reduce the torques even more than for flat surfaces. This shows that the approach is versatile and can be applied to a wide variety of problems.

Part IV

Inverse Kinematics of Robotic Manipulators with no Closed Form Solution

Chapter 12

Iterative Solutions to the Inverse Geometric Problem for Manipulators with no Closed Form Solution

12.1 Abstract

A set of new iterative solutions to the inverse geometric problem is presented. The approach is general and does not depend on intersecting axes or calculation of the Jacobian. The solution can be applied to any manipulator and is well suited for manipulators for which convergence is poor for conventional Jacobian-based iterative algorithms. For kinematically redundant manipulators, weights can be applied to each joint to introduce stiffness and for collision avoidance. The algorithm uses the unit quaternion to represent the position of each joint and calculates analytically the optimal position of the joint when only the respective joint is considered. This sub-problem is computationally very efficient due to the analytical solution. Several algorithms based on the solution of this sub-problem are presented. For difficult problems, for which the initial condition is far from a solution or the geometry of the manipulator makes the solution hard to reach, it is shown that the algorithm finds a solution fairly close to the solution in only a few iterations.

12.2 Introduction

In general, motion control is performed in operational space or joint space (Khalil and Dombre, 2002). Operational space control has the advantage that the end-effector position and orientation are given in the Cartesian space. For operational space control, the transformation from operational to joint space is obtained by solving the *inverse kinematic problem*, which finds the joint velocities from the desired end-effector velocities. Oper-

ational space control has many advantages and is fast to compute. A drawback is that it strongly depends on the inverse Jacobian and that the transformation from operational to joint space is performed inside the feedback loop so that the time-step of the controller strongly depends of the complexity of this transformation (Perdereau et al., 2002).

For joint space control, the transformation from operational space to joint space is obtained by solving the *inverse geometric problem*, i.e. to find the joint positions from the desired end-effector position/orientation. Then some joint space control scheme, independent of the task, can be designed. The disadvantage of this approach is that the inverse geometrics is a time-consuming problem to solve. The advantage is that the transformation from operational to joint space is moved outside the control loop. When kinematic redundancy is present, the inverse geometric approach also allows for optimising a general secondary criteria, and does not depend on finding a suitable inverse of the Jacobian, such as the Moore-Penrose generalised inverse, as for the inverse kinematic problem.

Another advantage of the inverse geometric approach is that each joint can be controlled more directly and given the desired characteristics such as joint stiffness, energy consumption, maximum velocity and obstacle avoidance. For the inverse Jacobian approach these characteristics must be added through the choice of the Jacobian. In some cases, such as the minimisation of energy through the Moore-Penrose, this is both efficient and elegant, but for other characteristics such an inverse Jacobian may be very hard or impossible to find.

Closed-form solutions to the inverse geometric problem are only known for certain types of robotic manipulators, so numerical approaches are widely studied and in many cases, such as for most redundant manipulators, represent the only solution to the problem. Numerical solutions are in general more time-consuming than closed-form solutions and are hence more suitable for off-line path planning. The results presented in this paper are based on the preliminary results presented in From and Gravdahl (2007a). Here the inverse geometric problem is treated as a pure optimisation problem. This allows the programmer to include a secondary objective which is used to give the manipulator motion the desired characteristics (Grudic and Lawrence, 1993; Wang and Chen, 1991; Luenberger, 2003). When redundancy is present, the redundant degrees of freedom are used to optimise this objective.

The novelty of the method presented is that the minimum of the cost function with respect to each joint is found analytically and this is exploited to develop a set of computationally efficient algorithms. The solution is shown for a cost function representing the position and orientation error of the end effector but can be expanded to include a general class of cost functions representing both global and local objectives.

Six algorithms are presented. The first three use *coordinate descent* which looks at one joint at the time. It is well known that the convergence of coordinate descent is slower than steepest descent and Newton's method. The advantage is that the analytic solution presented is a lot faster to solve than search algorithms in general. The last three methods can be looked upon as approximations of steepest descent where the gradient is estimated. It is argued that the step size can be set as a constant. Hence, an analytic and computationally efficient alternative to both the search direction and the step size of the steepest descent is presented.

It is shown that the algorithms that approximate the steepest descent have very good convergence and reliability for difficult problems. However, for easy problems, when the

initial guess is close to the solution, the convergence is better for conventional Jacobian-based algorithms than the algorithms presented in this paper. For problems for which the Jacobian-based algorithms have poor convergence or reliability, the algorithms presented are a better choice. A combination of the algorithms presented may give good and reliable performance for difficult problems but also reasonably good convergence close to the solution.

12.3 Representing Rotations

12.3.1 The Unit Quaternion

Any positive rotation ϕ about a fixed unit vector $\boldsymbol{n}$ can be represented by the quadruple (Kuipers, 2002)

$$Q = \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix}, \tag{12.1}$$

where $q_0 \in \mathbb{R}$ is known as the scalar part and $\boldsymbol{q} \in \mathbb{R}^3$ as the vector part. The unit quaternion $Q(\phi, \boldsymbol{n})$ is written in terms of ϕ and $\boldsymbol{n}$ by

$$q_0 = \cos(\frac{\phi}{2}), \qquad \boldsymbol{q} = \sin(\frac{\phi}{2})\boldsymbol{n}, \tag{12.2}$$

where $\boldsymbol{n}$ is unitary. Note that Q and $-Q$ represent the same rotation. This is referred to as the dual covering. The quaternion identity is given by $Q_I = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$. A multiplication of two quaternions is given by a quaternion product and is written in vector algebra notations as

$$P * Q = \begin{bmatrix} p_0 q_0 - \boldsymbol{p} \cdot \boldsymbol{q} \\ p_0 \boldsymbol{q} + q_0 \boldsymbol{p} + \boldsymbol{p} \times \boldsymbol{q} \end{bmatrix}. \tag{12.3}$$

Let $P = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix}^\mathsf{T}$ and $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$. A multiplication of two quaternions can then be written as the quaternion product as

$$P * Q = \begin{bmatrix} p_0 q_0 - p_1 q_1 - p_2 q_2 - p_3 q_3 \\ p_0 q_1 + p_1 q_0 + p_2 q_3 - p_3 q_2 \\ p_0 q_2 + p_2 q_0 + p_3 q_1 - p_1 q_3 \\ p_0 q_3 + p_3 q_0 + p_1 q_2 - p_2 q_1 \end{bmatrix}. \tag{12.4}$$

A pure quaternion is a quaternion with zero scalar part. Any vector, $\bar{\boldsymbol{v}} = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T}$ can be represented by a pure quaternion $\boldsymbol{v} = \begin{bmatrix} 0 & \bar{\boldsymbol{v}}^\mathsf{T} \end{bmatrix}^\mathsf{T}$. The conjugate of a quaternion is defined as

$$Q^* = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{bmatrix}^\mathsf{T}. \tag{12.5}$$

12.3.2 Quaternions and Rotations

Let a vector, $\bar{\boldsymbol{v}}_1$, be represented by the pure quaternion $\boldsymbol{v}_1$. This vector can be rotated the angle ϕ about the axis $\boldsymbol{n}$ by (Kuipers, 2002)

$$\boldsymbol{v}_2 = Q * \boldsymbol{v}_1 * Q^*. \tag{12.6}$$

Every vector $\bar{\boldsymbol{v}} \in \mathbb{R}^3$ can be represented by a pure quaternion, hence $\boldsymbol{v}$ is not necessarily a unit quaternion. The quaternion $Q(\phi, \boldsymbol{n})$ however is unitary. This represents the angle and the axis that the vector $\bar{\boldsymbol{v}}_1$ is to be rotated about. The resulting vector, $\bar{\boldsymbol{v}}_2$, is then of the same length as $\bar{\boldsymbol{v}}_1$ if and only if Q is a unit quaternion. The quaternion representation also leads to a useful formula for finding the shortest rotation from one orientation to another. Let P and Q be two orientations. Then, by taking

$$E = P^* * Q, \tag{12.7}$$

E will rotate P into Q by the shortest rotation. That is, E is the quaternion, out of all the quaternions that take P into Q, with the largest scalar part and thus the smallest angle.

Note that Equation (12.7) rotates one frame into another frame. By a *frame* is meant a coordinate system in $\mathbb{R}^3$ using Cartesian coordinates. One frame with respect to another frame represents three degrees of freedom and is referred to as an *orientation*. Equation (12.6) rotates one vector into another vector and has two degrees of freedom, in the same way as a point on a sphere can be defined by two coordinates. A unit vector with respect to a unit reference vector is referred to as a *direction*. Henceforth, when referred to direction, this is the direction of the z-axis of the body frame with respect to the z-axis of the reference frame, as the z-axis of the end effector is our main concern in this paper.

12.4 Quaternion Space Metric

The axis of a revolute joint, represented in the coordinate frame of the joint, is always constant. This is used in the following to simplify the computations. First the proximity of two frames is discussed, then this is applied to each joint to find the optimal position of the joint. By optimal position is meant the position of the joint that minimises the end effector orientation error, position error, or both.

There are many ways to represent the proximity or distance between two orientations (Yuan, 1988; Wen and Kreutz-Delgado, 1991; Hanson, 2006). One example which is proportional to the length of the geodesic path on the 4-dimensional unit sphere is

$$\Psi_r = \arccos(e_0) \tag{12.8}$$

where e_0 is taken from $E = P^* * Q$. The cost function in (12.8) can be identified with a physical property and is a metric function. The formal proof that (12.8) is a metric function is given in the Appendix. The geodesic path describes the shortest path from one orientation to another. Choosing that path on the 4-dimensional unit sphere gives a well-defined and computationally efficient metric.

A computationally more efficient cost function representing the rotational part is given simply by

$$\Psi_r = 1 - e_0. \tag{12.9}$$

This cost function lacks the property that it can be identified with a physical property directly and it is not a metric function. Also, its minimum is given by $e_0 = \pm 1$, for which the two orientations are identical, and the maximum is given by zero, for which the orientations point in the opposite directions. However, due to the light computational complexity, this cost function turns out to be very efficient.

A cost function on $SE(3)$ will depend on the weighing of the rotational and translational part. On its general form, it is given by

$$\Psi = w_t \Psi_t + w_r \Psi_r \tag{12.10}$$

where w_t and w_r are the weights, the translational part, Ψ_t is chosen as the standard Euclidean norm and the rotational part is the metric in (12.8)

$$\Psi = w_t \, ||p_0 - p_1|| + w_r \arccos(e_0). \tag{12.11}$$

or alternatively the cost function in (12.9)

$$\Psi = w_t \, ||p_0 - p_1|| + w_r (1 - e_0). \tag{12.12}$$

Definition 12.1 (Quaternion Space Proximity)**.** Given two orientations represented by the two quaternions P and Q. Let the *error quaternion* be denoted

$$E = P^* * Q. \tag{12.13}$$

Then the scalar part of E, e_0, describes the proximity of the two frames.

Definition 12.2 (Minimal Rotation)**.** The bigger (closer to 1)[1] the error quaternion scalar part e_0, the closer are the two orientations P and Q.

That this is a perfectly good description of the proximity of two frames even though it does not represent a physical property directly. The geodesic path can, however, be found through Equation (12.8) .

Consider the set of orientations for which the identity frame is rotated about the z-axis. The problem to find the orientation P_z from this set that is closest to some arbitrary orientation Q is considered.

Proposition 12.1 (Optimal Rotation)**.** *Consider an orientation* $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$. *The orientation described by the quaternion* $P_z = \begin{bmatrix} p_0 & 0 & 0 & p_3 \end{bmatrix}^T$ *that is closest to* Q *(by Definitions 12.1 and 12.2) is given by*

$$p_0 = \frac{\pm_s q_0}{\sqrt{q_0^2 + q_3^2}} \tag{12.14}$$

$$p_3 = \frac{\pm_s q_3}{\sqrt{q_0^2 + q_3^2}} \tag{12.15}$$

where the two $\pm_s$ *have the same sign.*

Proof. $E = P^* * Q$ can be written

$$\begin{bmatrix} e_0 \\ e_3 \end{bmatrix} = \begin{bmatrix} p_0 & p_3 \\ -p_3 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_3 \end{bmatrix} \tag{12.16}$$

[1] Note that an equally good description of proximity is given when e_0 approaches -1. As $\cos(\frac{\psi}{2})$ is positive for ψ in the chosen interval $(-\pi, \pi)$, the positive value of e_0 is chosen.

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} p_0 & p_3 \\ -p_3 & p_0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \tag{12.17}$$

By Definitions 12.1 and 12.2, the quaternion P_z that is closest to Q is found by the error quaternion with e_0 closest to 1.

$$e_0 = p_0 q_0 + p_3 q_3 \tag{12.18}$$

$$= q_0 \cos(\frac{\psi}{2}) + q_3 \sin(\frac{\psi}{2}). \tag{12.19}$$

$$\frac{\mathrm{d}e_0}{\mathrm{d}\psi_0} = -\frac{q_0}{2} \sin(\frac{\psi}{2}) + \frac{q_3}{2} \cos(\frac{\psi}{2}). \tag{12.20}$$

Let $\frac{\mathrm{d}e_0}{\mathrm{d}\psi_0} = 0$. Then

$$\tan(\frac{\psi}{2}) = \frac{q_3}{q_0}. \tag{12.21}$$

Then by using $\arctan(x) = \arcsin\left(\frac{x}{\sqrt{1+x^2}}\right)$ (Bronshtein et al., 2003), ψ is written as

$$\psi = 2 \arctan(\frac{q_3}{q_0}) \tag{12.22}$$

$$= 2 \arcsin\left(\frac{\frac{q_3}{q_0}}{\sqrt{1 + (\frac{q_3}{q_0})^2}}\right) \tag{12.23}$$

$$= 2 \arcsin\left(\frac{q_3}{\sqrt{q_0^2 + q_3^2}}\right). \tag{12.24}$$

From the definition of the quaternion

$$\psi = 2 \arcsin(p_3). \tag{12.25}$$

By comparing Equations (12.24) and (12.25), Equation (12.15) is given. Similarly, by $\arctan(x) = \arccos\left(\frac{1}{\sqrt{1+x^2}}\right) \mathrm{sgn}(x)$,

$$\psi = 2 \arctan(\frac{q_3}{q_0}) \tag{12.26}$$

$$= 2 \arccos\left(\frac{1}{\sqrt{1 + (\frac{q_3}{q_0})^2}}\right) \mathrm{sgn}(\frac{q_3}{q_0}) \tag{12.27}$$

$$= 2 \arccos\left(\frac{q_0}{\sqrt{q_0^2 + q_3^2}}\right) \mathrm{sgn}(\frac{q_3}{q_0}). \tag{12.28}$$

Note that the sign of $\psi = 2 \arccos(p_0)\mathrm{sgn}(\psi)$ is given by Equation (12.25). Hence, Equation (12.14) is found. For ψ to be in the interval $(-\pi, \pi)$, the sign $\pm_s$ is chosen so that e_0 is positive. □

Similar results are found when P rotates about the x- and y-axis. The largest rotation is given when e_0 is close to zero.

$$e_0 = p_0 q_0 + p_3 q_3 \tag{12.29}$$

$$= q_0 \cos(\frac{\psi}{2}) + q_3 \sin(\frac{\psi}{2}) = 0. \tag{12.30}$$

$$\tan(\frac{\psi}{2}) = -\frac{q_0}{q_3}. \tag{12.31}$$

Similar to the proof of Proposition 12.1, the orientation P_z furthest away from Q is given by

$$p_0 = \frac{\pm_s q_3}{\sqrt{q_0^2 + q_3^2}} \tag{12.32}$$

$$p_3 = \frac{\pm_t q_0}{\sqrt{q_0^2 + q_3^2}} \tag{12.33}$$

where the $\pm_s$ and $\pm_t$ have opposite signs.

12.5 Optimisation Algorithms

12.5.1 Descent Methods

This section presents some important approaches to solve a general optimisation problem by iterative algorithms (Luenberger, 2003).

Definition 12.3 (Descent Algorithm)**.** An algorithm that for every new point generated, decreases the corresponding value of some function, is called a descent algorithm.

If an algorithm is not descent, it is not guaranteed that the cost function decreases at every iteration. This property is desirable, but not required. Luenberger (2003) shows that the first order necessary condition is satisfied ($\nabla f = 0$) for descent algorithms. A similar proof cannot be given for algorithms that are not descent.

12.5.2 Steepest Descent

The most common method for the minimisation of a function of several variables is the steepest decent, or the gradient method. The steepest descent is given by the iterative algorithm

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k)^\mathsf{T} \tag{12.34}$$

where α^k is a non-negative scalar minimising $f(x^k - \alpha^k \nabla f(x^k))$. α^k is found by a search in the direction of the negative gradient for a minimum of this line. Convergence to a point where $\nabla f(x) = 0$ can be proven (Luenberger, 2003).

12.5.3 Coordinate Descent Methods

The coordinate descent algorithm optimises a given cost function $f(x)$, $x \in \mathbb{R}^n$, by sequentially minimising with respect to each of the components, x_i, for $i = 1 \ldots n$. The convergence of coordinate descent is in general poorer than the steepest descent. Coordinate descent is, however, easy to implement and, as the gradient is not required, a fast solution to the sub-problem makes these algorithms relatively fast.

12.5.4 Position and Orientation Error

This section presents a set of algorithms that solve the inverse geometric problem as seen from one joint. The solution of this sub-problem is the basis for all the algorithms presented in the next sections. Assume that only one joint can be moved, and consider the problem of finding the joint position which minimises the given cost function. All the algorithms presented are based on the analytical solution of a minimisation problem on $SE(3)$. This analytical solution guarantees that every sub-problem is computationally very efficient.

In the following, the principal cost function, representing the position and orientation error is presented. All cost functions presented are well-defined. If the cost function is extended to also include some secondary objective, this will depend on the task, and must be worked out in each case. The problem is solved for revolute joints only.

The algorithms in this section are based on two different optimisation problems. One with the position and orientation treated separately, and one where the cost function represents the sum of the position and orientation error. In this case the solution depends on the choice of units. As angles and lengths cannot simply be added together, care must be taken.

Position Cost Function

Let the desired position $P_d = \begin{bmatrix} 0 & x_d & y_d & z_d \end{bmatrix}^\mathsf{T}$ and current position $P_c = \begin{bmatrix} 0 & x_c & y_c & z_c \end{bmatrix}^\mathsf{T}$ of the end effector be given in the frame of joint i. Assume that the current position can be rotated about the z-axis, and hence represents one degree of freedom, given by all quaternions on the form $Q_z = \begin{bmatrix} \cos(\frac{\psi}{2}) & 0 & 0 & \sin(\frac{\psi}{2}) \end{bmatrix}^\mathsf{T}$ for $-\pi < \psi < \pi$. Then, the solution to the problem of finding the quaternion that takes P_c as close to P_d as possible is given by minimising

$$g_p(\psi) = (x_d - \hat{x}_c)^2 + (y_d - \hat{y}_c)^2 + (z_d - \hat{z}_c)^2, \tag{12.35}$$

where

$$\hat{P}_c = Q_z * P_c * Q_z^*. \tag{12.36}$$

By noting that

$$\hat{P}_c = \begin{bmatrix} 0 \\ \hat{x}_c \\ \hat{y}_c \\ \hat{z}_c \end{bmatrix} = \begin{bmatrix} 0 \\ x_c \cos\psi - y_c \sin\psi \\ y_c \cos\psi + x_c \sin\psi \\ z_c \end{bmatrix} \quad \text{for} - \pi < \psi < \pi, \tag{12.37}$$

$g_p(\psi)$ can be written as

$$g_p(\psi) = K_\psi + a_\psi \cos(\psi) + b_\psi \sin(\psi), \tag{12.38}$$

where

$$K_\psi = x_d^2 + y_d^2 + z_d^2 + x_c^2 + y_c^2 + z_c^2 - 2z_d z_c, \tag{12.39}$$
$$a_\psi = -2(x_d x_c + y_d y_c), \tag{12.40}$$
$$b_\psi = 2(x_d y_c - y_c x_d). \tag{12.41}$$

Similarly when the freedom is given about the y-axis, $g_p(\theta)$ is given by

$$g_p(\theta) = K_\theta + a_\theta \cos(\theta) + b_\theta \sin(\theta), \tag{12.42}$$

where

$$K_\theta = x_d^2 + y_d^2 + z_d^2 + x_c^2 + y_c^2 + z_c^2 - 2y_d y_c, \tag{12.43}$$
$$a_\theta = -2(x_d x_c + z_d z_c), \tag{12.44}$$
$$b_\theta = 2(z_d x_c - x_d z_d). \tag{12.45}$$

The rotation that minimises the position error of the end effector is given by setting $\frac{\mathrm{d}g_p(\psi)}{\mathrm{d}\psi} = 0$ and $\frac{\mathrm{d}g_p(\theta)}{\mathrm{d}\theta} = 0$:

$$\psi_{min} = \arctan 2\left(\frac{b_\psi}{a_\psi}\right) \pm \pi, \tag{12.46}$$
$$\theta_{min} = \arctan 2\left(\frac{b_\theta}{a_\theta}\right) \pm \pi \tag{12.47}$$

for a rotation about the z- and y-axes, respectively. In order to choose the solution that corresponds to the minimum and not the maximum value of g, choose the solution for which

$$\frac{\mathrm{d}^2 g_p(\theta)}{\mathrm{d}\theta^2} > 0. \tag{12.48}$$

Which solution to choose can also be determined by the following lemma.

Lemma 12.1. *Given a function $g(\theta)$ on the form*

$$g(\theta) = K + a\cos(\theta) + b\sin(\theta), \tag{12.49}$$

evaluated on $-\pi < \theta < \pi$. Let θ_{min} minimise $g(\theta)$. Then the following is always true

$$b > 0 \Rightarrow \theta_{min} < 0, \tag{12.50}$$
$$b < 0 \Rightarrow \theta_{min} > 0. \tag{12.51}$$

Proof. The lemma is proved by contradiction. Let θ_{min} minimise $g(\theta)$. Assume that $b > 0$ and $\theta_{min} > 0$. Then on the interval $-\pi < \theta < \pi$, we have that $a\cos(\theta_{min}) = a\cos(-\theta_{min})$ and $b\sin(\theta_{min}) > b\sin(-\theta_{min})$. Thus we have that $g(\theta_{min}) > g(-\theta_{min})$ which is a contradiction as θ_{min} was assumed to minimise $g(\theta)$. Similarly for $b < 0$. □

Orientation Cost Function

Similarly, the orientation error can be given by the difference between the desired orientation D and the current orientation C. Let D and C be given in the frame of joint i and let $\hat{C} = Q_z * C$ represent all the reachable orientations by rotating about the z-axis.

$$\hat{C} = Q_z * C = \begin{bmatrix} c_0 \cos(\frac{\psi}{2}) - c_3 \sin(\frac{\psi}{2}) \\ c_1 \cos(\frac{\psi}{2}) - c_2 \sin(\frac{\psi}{2}) \\ c_2 \cos(\frac{\psi}{2}) + c_1 \sin(\frac{\psi}{2}) \\ c_3 \cos(\frac{\psi}{2}) + c_0 \sin(\frac{\psi}{2}) \end{bmatrix}, \quad \text{for } -\pi < \psi < \pi \tag{12.52}$$

The orientation error is then given by

$$\begin{aligned} g_o(\psi) =& (d_0 - \hat{c}_0(\psi))^2 + (d_1 - \hat{c}_1(\psi))^2 + (d_2 - \hat{c}_2(\psi))^2 + (d_3 - \hat{c}_3(\psi))^2 \\ =& 2 - 2(c_0 d_0 + c_1 d_1 + c_2 d_2 + c_3 d_3) \cos(\frac{\psi}{2}) \\ &+ 2(c_3 d_0 + c_2 d_1 - c_1 d_2 - c_0 d_3) \sin(\frac{\psi}{2}). \end{aligned} \tag{12.53}$$

$g_o(\psi)$ can be written as

$$g_o(\psi) = K_\psi + c_\psi \cos(\frac{\psi}{2}) + d_\psi \sin(\frac{\psi}{2}), \tag{12.54}$$

where

$$K_\psi = 2, \tag{12.55}$$
$$c_\psi = -2(c_0 d_0 + c_1 d_1 + c_2 d_2 + c_3 d_3), \tag{12.56}$$
$$d_\psi = 2(c_3 d_0 + c_2 d_1 - c_1 d_2 - c_0 d_3). \tag{12.57}$$

Similarly when the y-axis is the revolute axis.

$$\begin{aligned} g_o(\theta) =& 2 - 2(c_0 d_0 + c_1 d_1 + c_2 d_2 + c_3 d_3) \cos(\frac{\theta}{2}) \\ &+ 2(c_2 d_0 - c_3 d_1 - c_0 d_2 + c_1 d_3) \sin(\frac{\theta}{2}). \end{aligned} \tag{12.58}$$

$g_o(\theta)$ can then be written as

$$g_o(\theta) = K_\theta + c_\theta \cos(\frac{\theta}{2}) + d_\theta \sin(\frac{\theta}{2}), \tag{12.59}$$

where

$$K_\theta = 2, \tag{12.60}$$
$$c_\theta = -2(c_0 d_0 + c_1 d_1 + c_2 d_2 + c_3 d_3), \tag{12.61}$$
$$d_\theta = 2(c_3 d_0 + c_2 d_1 - c_1 d_2 - c_0 d_3). \tag{12.62}$$

The advantage of this approach is that the cost function can be used as an error measure directly. The quaternion representation also allows the optimal rotation to be computed somewhat faster, but then the error function needs to be calculated separately as in Johnson (1995) and From (2006).

12.5.5 Orientation and Position Cost Function

The total position and orientation error can be given by $g(\psi) = g_p(\psi) + g_o(\psi)$. $g_p(\psi)$ and $g_o(\psi)$ are taken from Equations (12.38) and (12.54), respectively, so that the minimum of $g(\psi)$ is given by

$$\frac{\mathrm{d}g(\psi)}{\mathrm{d}\psi} = 0 \tag{12.63}$$

where

$$\frac{\mathrm{d}g(\psi)}{\mathrm{d}\psi} = b_\psi \cos(\psi) + d_\psi \cos(\frac{\psi}{2}) - a_\psi \sin(\psi) - c_\psi \sin(\frac{\psi}{2}). \tag{12.64}$$

This can be turned into an equation of degree four which can be solved analytically, for example by Ferrari's method. This will give four solutions. The solution that results in the smallest value of $g(\psi)$ is then chosen.

However, by avoiding the half angles in Equation (12.64), the solution is found simply by the inverse tangent and the computational complexity is reduced. In Wang and Chen (1991) a ψ is found by maximising $g(\psi)$. In the following, a cost function, representing the sum of the position and orientation error is presented. This cost function can then be used as a threshold limit directly, which was not the case in Wang and Chen (1991). The approach resembles the one in Ahuactzin and Gupka (1999), but allows the programmer to weigh the importance of the position and orientation error.

The cost function can be written as a function of ψ by representing the desired orientation of each joint by a rotation of the three unit vectors by ${}^{x}Q_d = Q_d * \boldsymbol{e}_i * Q_e^*$. ${}^{y}Q_d$ and ${}^{z}Q_d$ are constructed similarly from $\boldsymbol{e}_j$ and $\boldsymbol{e}_k$ where $\boldsymbol{e}_i, \boldsymbol{e}_j, \boldsymbol{e}_k$ are the unitary axes. Then the unitary axes are transformed by the quaternion Q_d into

$${}^{x}Q_d = Q_d * \boldsymbol{e}_i * Q_d^* = \begin{bmatrix} 0 \\ q_0^2 + q_1^2 - q_2^2 - q_3^2 \\ 2(q_1 q_2 + q_0 q_3) \\ 2(q_1 q_3 - q_0 q_2) \end{bmatrix}, \tag{12.65}$$

$${}^{y}Q_d = Q_d * \boldsymbol{e}_j * Q_d^* = \begin{bmatrix} 0 \\ 2(q_1 q_2 - q_0 q_3) \\ q_0^2 - q_1^2 + q_2^2 - q_3^2 \\ 2(q_0 q_1 + q_2 q_3) \end{bmatrix}, \tag{12.66}$$

$${}^{z}Q_d = Q_d * \boldsymbol{e}_k * Q_d^* = \begin{bmatrix} 0 \\ 2(q_0 q_2 + q_1 q_3) \\ 2(q_2 q_3 - q_0 q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{12.67}$$

Then the cost function can be written as

$$g(\psi) = w_p g_p(\psi) + w_o g_o(\psi) \tag{12.68}$$

where w_p and w_o are constant weights, $g_p(\psi)$ is given by Equation (12.38) and $g_p(\psi)$ is found similarly by representing the difference between the desired position of the unitary axes and the current position of the same axes. The desired position for the x-axis is given by ${}^xQ_d = \begin{bmatrix} 0 & {}^xx_d & {}^xy_d & {}^xz_d \end{bmatrix}$. Assume that the z-axis is the revolute axis. Then the position of the unitary x-axis is given by ${}^xQ_c = \begin{bmatrix} 0 & \cos(\psi) & \sin(\psi) & 0 \end{bmatrix}$ and the difference is written as

$$\begin{aligned} {}^xg_o(\psi) &= ({}^xx_d - \cos(\psi))^2 + ({}^xy_d - \sin(\psi))^2 + ({}^xz_d - 0)^2 \\ &= 2 - 2{}^xx_d \cos(\psi) - 2{}^xy_d \sin(\psi), \end{aligned} \tag{12.69}$$

and similarly for the y- and z-axes. By adding these three to Equation (12.38), $g(\psi)$ can be written as

$$\begin{aligned} g(\psi) &= w_p g_p(\psi) + w_o({}^xg_o(\psi) + {}^yg_o(\psi) + {}^zg_o(\psi)) \\ &= K_\psi + a_\psi \cos(\psi) + b_\psi \sin(\psi) \end{aligned} \tag{12.70}$$

where

$$\begin{aligned} K_\psi &= w_p \left(x_d^2 + y_d^2 + z_d^2 + x_c^2 + y_c^2 + z_c^2 - 2z_d z_c\right) + w_o (6 - 2{}^zz_d), \\ a_\psi &= -2w_p(x_d x_c + y_d y_c) - 2w_o({}^xx_d + {}^yy_d), \\ b_\psi &= 2w_p(x_d y_c - y_d x_c) + 2w_o({}^yx_d - {}^xy_d). \end{aligned}$$

Similarly when the y-axis is the revolute axis

$$g(\theta) = K_\theta + a_\theta \cos(\theta) + b_\theta \sin(\theta) \tag{12.71}$$

where

$$\begin{aligned} K_\theta &= w_p \left(x_d^2 + y_d^2 + z_d^2 + x_c^2 + y_c^2 + z_c^2 - 2y_d y_c\right) + w_o (6 - 2{}^yy_d), \\ a_\theta &= -2w_p(x_d x_c + z_d z_c) - 2w_o({}^xx_d + {}^zz_d), \\ b_\theta &= 2w_p(z_d x_c - x_d z_c) + 2w_o({}^xz_d - {}^zx_d). \end{aligned}$$

The minimum of the cost function, with respect to each joint, is given by Equation (12.46) and the error is given by $E = K + a$ (set $\psi = \theta = 0$ in (12.70) and (12.71)).

For redundant manipulators, the cost function can be expanded to include an addition term

$$g(\psi) = w_p g_p(\psi) + w_o g_o(\psi) + w_r g_r(\psi). \tag{12.72}$$

Whenever g_r can be written on the form of (12.70) the same analytical solution to the sub-problem can be found. This is a large class of cost functions that allows a great variety of secondary objectives to be included in the cost function, such as distance to obstacles and elbow position.

Note also that for the pointing task, Equation (12.70) reduces to

$$g(\psi) = w_p g_p(\psi) + w_o{}^zg_o(\psi) \tag{12.73}$$

which is widely used in applications such as spray painting, welding and high pressure water jets. In this case only the direction of the end-effector tool is considered and thus the computational complexity is reduced.

12.6 Solutions to the Inverse Geometric Problem

12.6.1 Algorithm 1 - Coordinate Descent

The coordinate descent algorithm optimises a cost function with respect to each of the variables of the cost function (Wang and Chen, 1991). That is, for each joint in the chain, the minimum of the cost function, when only the respective joint is moved, is found.

There are several different ways the algorithm can work its way through the chain:

- Start from the end and work its way towards the base.
- Start from the base and work its way towards the end.
- Start from one end and sweep its way towards the other and then back (Aitken double sweep method).
- If the gradient is known, select the coordinate (in this case the joint) that corresponds to the largest (in absolute value) component of the gradient vector (Gauss-Southwell Method, presented in the Section 12.6.2).

The cost function must be objective, i.e. independent of the coordinate frame in which it is measured (Lin and Burdick, 2000), and preferably describing some physical property, as the sum of the position and orientation error. Objectivity is important because all the calculations are done in local coordinates, and thus the coordinate frame changes for each iteration. Objectivity is in this case sufficient to guarantee that the algorithm is descent and convergent (to a point satisfying the first order necessary condition). The cost function should also be computationally efficient, i.e. the minimum of the cost function should be found analytically.

The cost function presented in Section 12.5.5 have these properties. This cost function, together with an algorithm that starts from the end and moves its way towards the base, is a fast and stable algorithm. The cost functions representing rotation or orientation error only are also well-defined on $SO(3)$ and $\mathbb{R}^3$, respectively. They can also be combined as described in Section 12.5.5 to a metric function on $SE(3)$. Caution must be taken when dealing with metrics on $SE(3)$, as it will depend on the choice of units and an unfortunate implementation of the algorithm may cause the algorithm to fail to converge. This is, as will be clear in the following, for example the case when iteratively optimising with respect to orientation and position error.

Three different approaches are presented:

Algorithm 1a: Loop until the error is under a threshold limit or a maximum number of iterations is performed.

- for each joint, in a pre-defined order, find the joint position that locally minimises the position error of the end-effector, as in Section 12.5.4.
- for each joint, in a pre-defined order, find the joint position that locally minimises the orientation error of the end-effector, as in Section 12.5.4.

Algorithm 1b: Loop until the error is under a threshold limit or a maximum number of iterations is performed. For each joint, in a pre-defined order

- find the joint position that locally minimises the position error of the end-effector, as in Section 12.5.4.
- find the joint position that locally minimises the orientation error of the end-effector, as in Section 12.5.4.

Algorithm 1c: Loop until the error is under a threshold limit or a maximum number of iterations is performed. For each joint, in a pre-defined order

- Minimise a cost function representing the sum of the position and orientation error, as in Section 12.5.5.

The change of reference frame on the cost function must be studied. The cost function needs to be objective, as defined in Lin and Burdick (2000), if not, the algorithm may fail to converge. A well defined metric function will guarantee that the the value of the cost function does not change with the change of reference frame which again means that it does not change with the joint. The cost function must also be so that the total error decreases when iterating between $SO(3)$ and $\mathbb{R}^3$ such as in Algorithms 1a and 1b. This is not guaranteed by just successively iterating between position and orientation as a decrease in the orientation error might cause an increase in the position error and vice versa. There is no guarantee that the total error decreases for every iteration.

12.6.2 Algorithm 2 - Modified Gauss-Southwell

The Gauss-Southwell Method determines the largest component of the gradient $\nabla g(x)$ and chooses this for descent. This sub-section presents an alternative approach, where the minimum of the cost function is found for each joint. The joint that corresponds to the smallest possible value of the cost function is then chosen. This is found simply by Equation (12.46). This approach is computationally more efficient than to compute the gradient. It will also converge faster (at least in the beginning) because the joint that corresponds to the maximum possible decrease of the cost function is always chosen. This algorithm is descent.

12.6.3 Algorithm 3 - Gauss-Southwell

The method presented above can be modified somewhat so that each joint is chosen by the steepest descent instead of maximum possible descent. Assume that the position of each joint that results in the minimum of the cost function $g(x)$ is found. Denote this by $\hat{x}_i^k$ for joint i and iteration k. The *rate* of decrease with respect to this joint is estimated by

$$\frac{\partial g(x_i^k)}{\partial x_i^k} \approx \frac{g(\hat{x}_i^k) - g(x_i^k)}{\left|\hat{x}_i^k - x_i^k\right|}, \qquad \text{for } i = 1 \ldots n. \tag{12.74}$$

This is a good estimate only when $\left|\hat{x}_i^k - x_i^k\right|$ is small. Then the joint with the largest corresponding absolute value of the "gradient" is chosen. This approach is different from

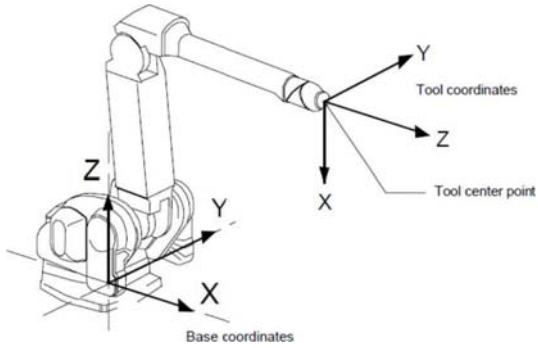


Figure 12.1: General structure of a robotic manipulator.

the solution given in Section 12.6.2 in that not only the absolute minimum is taken into account, but also how much the manipulator has to move reflects the choice of search direction, which leads to a more energy preserving solution. The joint update is then given by

$$x_i^{k+1} = x_i^k + w_i(\hat{x}_i^k - x_i^k), \qquad \text{with } 0 < w_i \leq 1. \tag{12.75}$$

12.6.4 Algorithm 4 - Steepest Descent

Equation (12.74) gives information about all the joints. This information can be exploited by applying (12.75) to all the joints for every iteration. As the optimal position of each joint is found, assuming that all the other joints are fixed, the weights $0 < w_i \leq 1$ need to be chosen conservatively. As this approach requires approximately the same computational burden as the approach in the previous section but all joints are moved, the convergence can be improved substantially. The algorithm is not descent, and convergence cannot be proven. This is due to the fact that Equation (12.74) is an estimate of the gradient and not the actual gradient. In some cases the w_i's must be chosen very small which makes the convergence very slow.

12.6.5 Algorithm 5 - Manipulator Dependent Steepest Descent

The manipulator structure can be taken into account to improve convergence. For instance if two joints work in the same "direction" in the operational space, they should be scaled down so that the sum of the two joints will result in the desired movement, and not each one looked at separately. By studying the structure of the manipulator in Figure 12.1, joint 1 is seen to be very much decoupled from the others when it comes to the effect on the end-effector position and orientation, and thus x_1^{k+1} is set close to $\hat{x}_1^k$. Joint 2 and 3, however, are strongly coupled, so $w_{2,3}$ should be set to about 0,5. The three wrist joints should also be scaled due to coupling. In addition, this scaling vector should be scaled down somewhat by a factor $0 < w_s \leq 1$, to ensure convergence. The following scaling vector is suggested for a manipulator with a structure similar to the one in Figure 12.1:

$$W = w_s \begin{bmatrix} 1 & 0.5 & 0.5 & 0.3 & 0.3 & 0.3 \end{bmatrix}. \tag{12.76}$$

As the previous algorithm, this algorithm is not descent. However, w_s can be set so that the behaviour of the algorithm resembles that of a descent algorithm. This is done at the cost of fast convergence. A simple approach to make the algorithm behave like a descent algorithm is to perform a test for every iteration to check whether the cost function has decreased or not. Then, if it has not, w_s should be reduced until a decrease in the cost function is obtained. For the steepest descent, a decrease of the cost function can be guaranteed as $w \to 0$. As $\nabla g(x)$ is only an approximation of the gradient, this cannot be guaranteed in this case.

12.6.6 Algorithm 6 - Steepest Descent with Gradient Estimate

Equation (12.74) can also be used to make an estimate of the gradient of the cost function. If the absolute sign is removed, the gradient of $g(x^k)$ can be estimated as

$$\widehat{\nabla} g(x^k) \approx \begin{bmatrix} \frac{g(\hat{x}_1^k) - g(x_1^k)}{\hat{x}_1^k - x_1^k} \\ \vdots \\ \frac{g(\hat{x}_n^k) - g(x_n^k)}{\hat{x}_n^k - x_n^k} \end{bmatrix} \tag{12.77}$$

As $g(x)$ is on the form of (12.70), $\left|\widehat{\nabla} g(x_i^k)\right| \leq \left|\nabla g(x_i^k)\right|$ for all i so that $\widehat{\nabla} g(x)$ is a conservative estimate of $\nabla g(x)$.

Now, Equation (12.77) can be applied to Equation (12.34) directly. The "step size" can be set similar to Equation (12.76) with (somewhat conservatively) $w_s = \min_{i=1\ldots n} \left|\hat{x}_i^k - x_i^k\right|$. When the solution approaches zero, the it can be simplified to $w_s = \left|\hat{x}_1^k - x_1^k\right|$.

It should be noted that when Equation (12.75) is applied to all joints, or the estimate of the gradient is applied in Equation (12.34), the algorithm is not descent. Again, however, the behaviour of the algorithm can be made descent by choosing the weights conservatively.

The steepest descent with gradient estimate differs from Algorithm 4 steepest descent in that for the steepest descent the optimal solution for each joint looked at separately is found, and then the update is done for all joints. For the steepest descent with the gradient estimate is the well known steepest descent method, but with an estimate of the gradient.

12.7 Numerical Examples

All the inverse geometric algorithms have been tested for a great variety of problems with the cost functions given in Sections 12.5.4 and 12.5.5. For comparison, the same test has also been done for a Jacobian-based inverse geometric algorithm. The Jacobian-based algorithm used in the simulations is an iterative algorithm based on the pseudo-inverse of the manipulator Jacobian, as the one presented in Robotics Toolbox (Corke, 1996). The convergence of the algorithms are tested for very difficult problems and very easy problems. Difficult problems are problems for which the solution is very far from the initial guess or the geometric considerations makes it difficult to "move" the manipulator from the initial condition to the solutions. For the easy problems the initial guess is chosen close to one of the solutions. 20 difficult and 20 easy problems are chosen and convergence

is investigated for the two cases for all algorithms presented. The convergence for easy and difficult problems for all algorithms presented are plotted with respect to iterations and time in Figures 12.4-12.7.

12.7.1 Algorithm 1 - Coordinate Descent

The conventional CCD presented in Section 12.6.1 is computationally fast. The convergence of the CCD algorithms can be found in Figures 12.2-12.3. The following algorithms are tested:

- Alg1a $(6 \rightarrow 1)$
- Alg1b $(6 \rightarrow 1)$
- Alg1c $(6 \rightarrow 1)$
- Alg1c $(1 \rightarrow 6)$
- Alg1c (double sweep)

where Alg1x refers to the algorithms in Section 12.6.1 and $(6 \rightarrow 1)$ means that the algorithm works its way through the chain from the end effector to the base.

It is clear that the first two algorithms that optimise iteratively between orientation and rotation error are not descent and convergence is poor. It is found that optimising with respect to one criteria, while disregarding the other, will not necessarily decrease the sum of the two cost functions.

The three algorithms presented that are based on a cost function representing the sum of the orientation and position error are all descent algorithms and convergence is reasonably good due to the analytical solution of each sub-problem. The algorithm that starts at the base and works its way towards the end of the chain has fastest convergence in the beginning an also finds the most accurate solution. The fast analytical solution to the sub-problem, presented in Section 12.5.5 makes this algorithm reasonably good. Alg1c $(6 \rightarrow 1)$ is chosen to compare convergence in Figures 12.4-12.7.

12.7.2 Algorithm 2 - Modified Gauss-Southwell

Gauss-Southwell is computationally slower as it finds the minimum for all the joints but only one joint is chosen for decrease. As the Modified Gauss-Southwell finds the minimum possible value of the cost function by moving one joint only, it has the best convergence in the beginning among the algorithms that move only one joint at the time. This makes this algorithm a very good choice when an approximate solution to the problem is needed. Convergence is very good for 5-10 iterations. After this the convergence flattens out and one should switch to another algorithm to find an exact solution.

12.7.3 Algorithm 3 - Gauss-Southwell

Also the Gauss-Southwell has good convergence in the beginning, but only for about 5 iterations. Then it flattens out and the closest solution found is farther from the desired

solution than for the Modified Gauss-Southwell. The algorithm easily gets stuck, and for the majority of the problems, it does not converge toward a correct solution. The algorithm can only be said to perform satisfactory for the first few iterations.

12.7.4 Algorithm 4 - Steepest Descent

The Steepest Descent moves all joints for every iteration which results in very good convergence. For a weight $w \approx 0.5$, the behaviour of the algorithm is very stable and a very accurate solution is found reasonably fast. This is the algorithm presented that best competes with the Jacobian approach when the initial condition is close to a solution. Also for more difficult problems, this is the algorithm that finds the most accurate solution if many iterations are allowed.

12.7.5 Algorithm 5 - Manipulator Dependent Steepest Descent

The convergence of the Manipulator Dependent Steepest Descent is about the same as the Steepest Descent, but convergence is better in the beginning for difficult problems. An algorithm that applies a few (5-10) iterations of the Manipulator Dependent Steepest Descent and then changes to Steepest Descent will give a fast and reliable algorithm which is easy to implement as the two algorithms are almost equal when it comes to implementation.

12.7.6 Algorithm 6 - Steepest Descent with Gradient Estimate

The Steepest Descent with Gradient Estimate is hard to tune and the weights need to be chosen relatively small for the algorithm to behave stable. This results poor convergence. The convergence is about the same as the Coordinate Descent methods, but the computational complexity makes this algorithm slower. The weight used in the simulations was $w_s = 0.05$.

12.7.7 Iteration Speed

The simulations were performed on an 2GHz processor. Table 12.1 shows the iteration speed of each algorithm. For algorithms 1-6, this is the time needed to analytically solve the optimisation problem and to update the joint position and the value of the objective function.

12.8 Conclusions

A new class of solutions to the inverse geometric problem is presented. Convergence is found to be very good for problems which cannot be solved efficiently or cannot be solved at all with Jacobian-based algorithms. For all tests, an approximate solution was found in only a few iterations. The analytical solution to the sub-problem guarantees computational efficiency. A combination of the algorithms presented will give a stable and fast solution to any inverse geometrics problem. For problems with initial condition close to a solution, conventional Jacobian-based algorithms converge faster. The algorithms presented are

Algoritm	Iteration Speed [ms]
Alg0	3.85
Alg1	2.62
Alg2	12.36
Alg3	12.46
Alg4	18.87
Alg5	18.88
Alg6	15.27

Table 12.1: Iteration speed of each algorithm

thus well suited to find an initial condition for the Jacobian-based algorithms in order to improve convergence and guarantee that a solution is found.

Appendix I

Formal Metric Proof

A metric on a set X is a function

$$\Psi : X \times X \to \mathbb{R} \tag{12.78}$$

which for all $x, y, z \in X$ satisfy the following conditions

1. $\Psi(x, y) \geq 0$
2. $\Psi(x, y) = 0$ if and only if $x = y$
3. $\Psi(x, y) = \Psi(y, x)$
4. $\Psi(x, z) \leq \Psi(x, y) + \Psi(y, z)$

Let $\mathbb{U}$ define the set of all quaternions of unit length

$$\mathbb{U}=\{(q_0, q_1, q_2, q_3)|q_0, q_1, q_2, q_3 \in \mathbb{R},\ q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1\} \tag{12.79}$$

Further let e_0 be the scalar part of $E = P * Q^*$ given by

$$e_0 = p_0 q_0 + p_1 q_1 + p_2 q_2 + p_3 q_3. \tag{12.80}$$

We will, without loss of generality, assume that all angles are in the interval $-\pi \leq \phi \leq \pi$.

Proposition 12.2. *The function*

$$\Psi_r = \mathbb{U} \times \mathbb{U} \to \mathbb{R} \tag{12.81}$$

given by $\Psi_r = \arccos(e_0)$*, is a metric function.*

Proof. For all $P, Q, R \in \mathbb{U}$ we have

1. $\Psi(P,Q) \geq 0$
   We have
$$-1 \leq e_0 \leq 1 \Rightarrow \arccos(e_0) \geq 0. \tag{12.82}$$

2. $\Psi(P,Q) = 0$ if and only if $P = Q$
   We have
$$\arccos(e_0) = 0 \tag{12.83}$$
   if and only if
$$e_0 = 1 \tag{12.84}$$
   for which $P = Q$.

3. $\Psi(P,Q) = \Psi(Q,P)$
$$\begin{aligned}\Psi(P,Q) &= \arccos(p_0q_0 + p_1q_1 + p_2q_2 + p_3q_3) \\ &= \arccos(q_0p_0 + q_1p_1 + q_2p_2 + q_3p_3) = \Psi(Q,P).\end{aligned} \tag{12.85}$$

4. $\Psi(P,R) \leq \Psi(P,Q) + \Psi(Q,R)$
   By definition the rotation $E = P * R^*$ takes P into R by the shortest rotation. This is obtained by the rotation
$$\phi_{PR} = 2\arccos\left(e_0^{PR}\right) \tag{12.86}$$
   where e_0^{PR} is the scalar part of $P * Q^*$. Thus we have that
$$\phi_{PR} \leq \phi_{PQ} + \phi_{QR}. \tag{12.87}$$
   Because the rotation from P to Q followed by the rotation from Q to R also take P into R, and from (12.86) and (12.87) we have
$$\begin{aligned}\phi_{PR} &\leq \phi_{PQ} + \phi_{QR} \\ \frac{1}{2}\phi_{PR} &\leq \frac{1}{2}\phi_{PQ} + \frac{1}{2}\phi_{QR} \\ \arccos\left(e_0^{PR}\right) &\leq \arccos\left(e_0^{PQ}\right) + \arccos\left(e_0^{QR}\right) \\ \Psi(P,R) &\leq \Psi(P,Q) + \Psi(Q,R)\end{aligned} \tag{12.88}$$
   which concludes the proof.

□

Finally we show, by contradiction that the function
$$\Psi_r = \mathbb{U} \times \mathbb{U} \to \mathbb{R} \tag{12.89}$$
given by $\Psi_r = 1 - e_0$, is *not* a metric function.

Given the triangular inequality

$$\begin{aligned}
\Psi(P,R) &\leq \Psi(P,Q) + \Psi(Q,R)) \\
(1 - e_0^{PR}) &\leq (1 - e_0^{PQ}) + (1 - e_0^{QR}) \\
-e_0^{PR} &\leq -(e_0^{PQ} + e_0^{QR} - 1) \\
e_0^{PR} &\geq e_0^{PQ} + e_0^{QR} - 1
\end{aligned} \tag{12.90}$$

Consider the following rotations

$$\begin{aligned}
P &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}} \\
Q &= \begin{bmatrix} \cos\left(\frac{\phi}{2}\right) & 0 & \sin\left(\frac{\phi}{2}\right) & 0 \end{bmatrix}^{\mathsf{T}} \\
R &= \begin{bmatrix} \cos\left(\frac{2\phi}{2}\right) & 0 & \sin\left(\frac{2\phi}{2}\right) & 0 \end{bmatrix}^{\mathsf{T}}
\end{aligned} \tag{12.91}$$

Then we have that both $P * R^*$ and $P * Q^*$ followed by $Q * R^*$ will take P into R. If we set $\phi = 0.1$ we have

$$\begin{aligned}
e_0^{PQ} &= 0.9988 \\
e_0^{QR} &= 0.9988 \\
e_0^{PR} &= 0.9950
\end{aligned} \tag{12.92}$$

and we have

$$\begin{aligned}
0.9950 &\geq 0.9988 + 0.9988 - 1 \\
0.9950 &\geq 0.9975
\end{aligned} \tag{12.93}$$

and thus a contradiction.

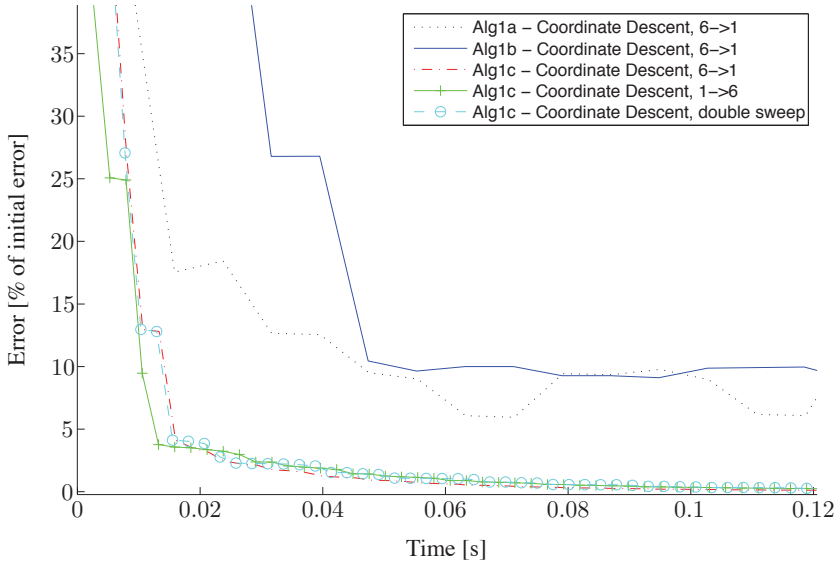


Figure 12.2: Convergence of Coordinate Cyclic Descent Algorithms that move one joint at the time. Initial conditions is set far from a solution.

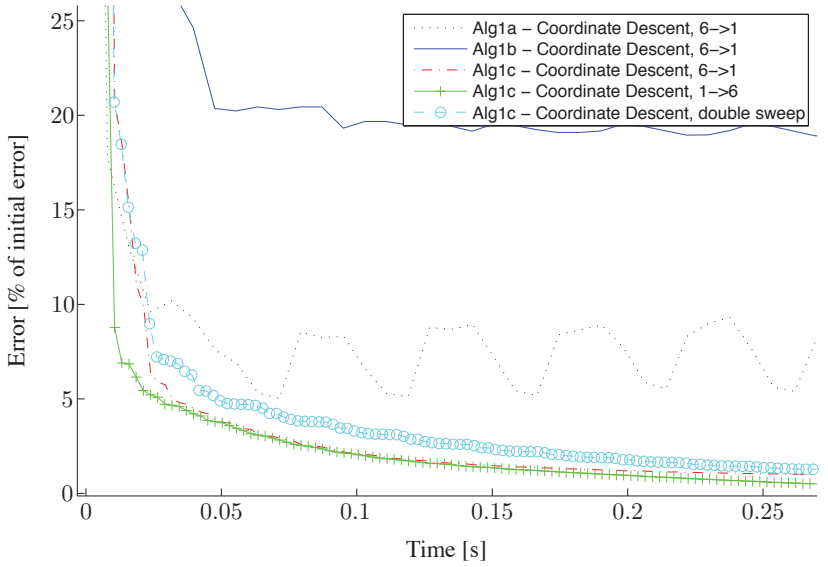


Figure 12.3: Convergence of Coordinate Cyclic Descent Algorithms that move one joint at the time. Initial conditions is set close to a solution.

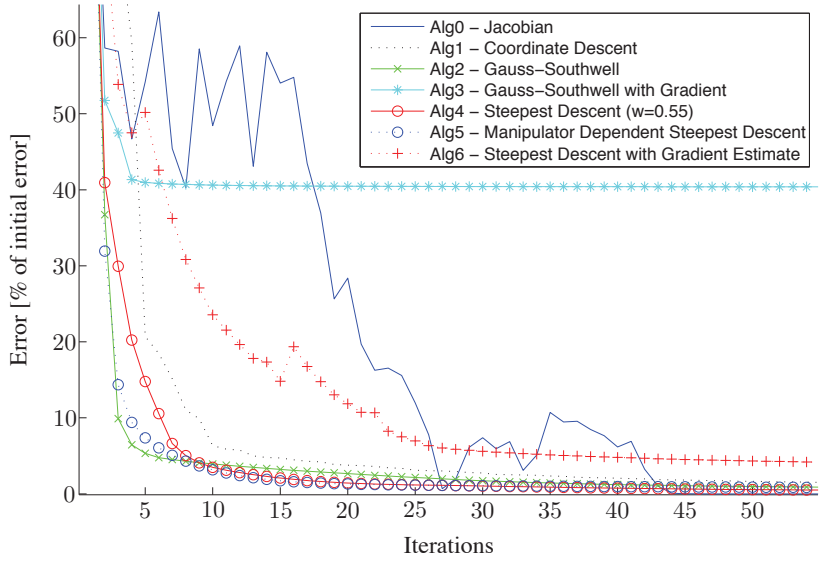


Figure 12.4: Convergence of algorithms with initial conditions far from a solution.

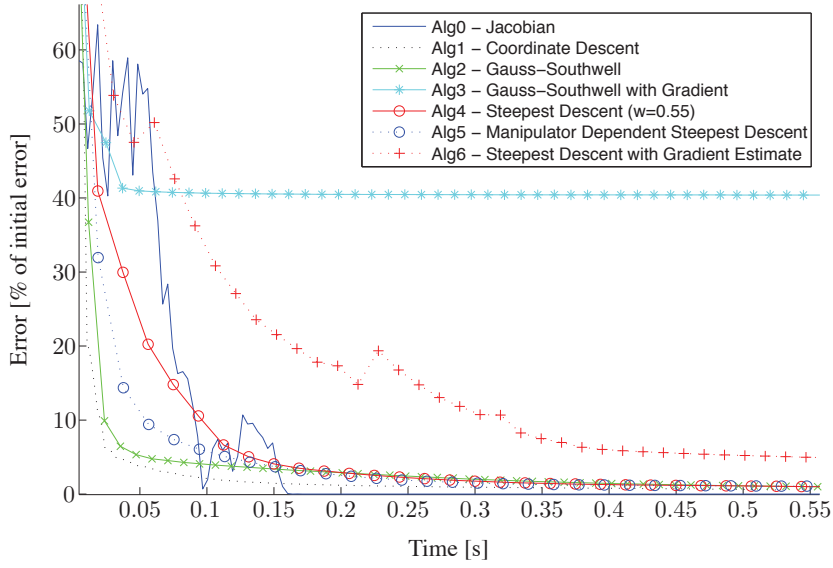


Figure 12.5: Convergence of algorithms with initial conditions far from a solution with respect to time.

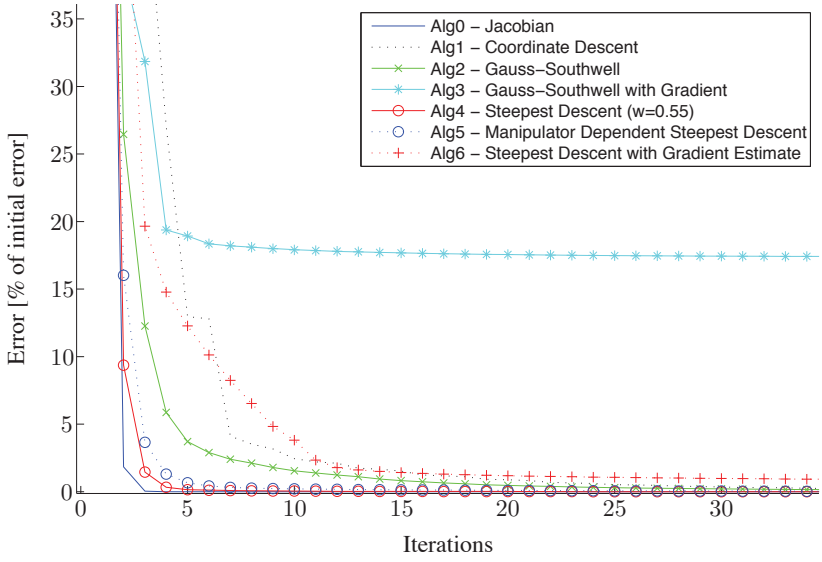


Figure 12.6: Convergence of algorithms with initial conditions close to solution.

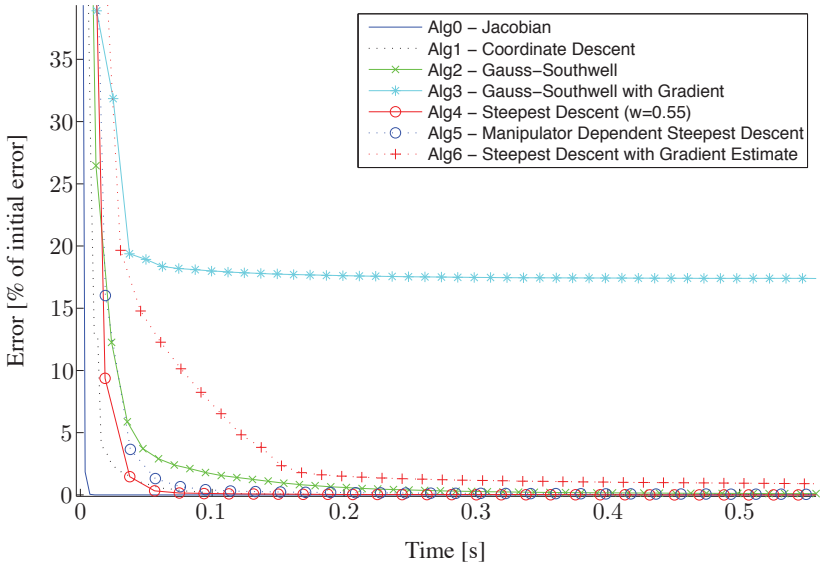


Figure 12.7: Convergence of algorithms with initial conditions close to solution with respect to time.

Part V

Concluding Remarks

Chapter 13

Conclusion

In this thesis we have addressed several issues that need to be solved before autonomous operation of off-shore oil and gas fields can see the light of day. The diversity in topics covered illustrates well how the transition from humanly operated platforms to autonomous operation requires advances in several different fields of research related to robotics. The problems that need to be solved are not only related to several different research areas: the solutions also need to represent considerable advances when it comes to robustness and efficiency, compared to today's technology. The transition to complete autonomous operation of off-shore platforms thus requires considerable effort in a wide variety of research areas and there is a need for considerable advances in almost all these areas before this transition can take place.

Today there are several processes that are automated on oil platforms. These are, however, mainly internal processes such as the control and operation of the "factory floor". The external processes, such as observation and maintenance, have not yet reached this level of automation. The main topic of this thesis is how to close this gap so that also the external processes can be automated.

Closing this gap will require advances also in areas not directly related to robotics. One example is a fundamental change to the platform design. Because tasks such as observation and maintenance up until now have been performed by humans, existing oil platforms are not built for robotic operation. The oil platforms of the future will thus be quite different from what we see today. Designing the next generation platforms is currently a very active research area and the final product will depend heavily on the different types of robotic solutions we choose. The platforms need to be constructed so that the robots can move around freely and have access to every part of the platform. The platforms will also be considerably smaller as the living facilities are removed. Platform design has not been treated in this dissertation, but the robotic solutions discussed will certainly affect the design of the platforms of the future.

In this thesis we have introduced some problems that we believe need to be addressed in order to obtain autonomous operation. For the topics addressed there are still a lot of work that needs to be done. In this sense we have not presented the final solution or the "right answer" to these problems and we have certainly not addressed all the topics that need to be addressed. We do believe, however, that we have addressed some important

topics and that we have made some contributions—some small and in some cases bigger—to these topics.

We have put considerable effort into making the theory presented as robust as possible. We believe this to be one of the main challenges when operating platforms located in remote areas. This is also one of the most difficult topics in robotics—maybe in any kind of engineering. Solutions need to work, if not, no-one will invest in them. Also, the systems need to work when unforeseen events occur. This is considerably more challenging than making solutions that work in structured environments. This is a very active research area that has been addressed in great depth in this thesis.

The first part of the thesis discussed robust mathematical representations of the dynamics of vehicle-manipulator systems. These models are used in simulation and control of the robots where a mathematically robust representation is important. We have presented the dynamic equations of these systems without the presence of singularities and in a computationally efficient way. This makes the proposed approach well suited for model-based control and implementation in simulation software. We have also presented solutions for robust operation of parallel or cooperating robots. The approach can be used in the design process, where we have shown how to design parallel manipulators that are fault tolerant with respect to torque failure. The approach can also be used during operation if a joint failure occurs in order to prevent damage to the surrounding robots or equipment.

Another aspect of developing solutions that the oil companies will invest in is to make the systems economically viable. This requires efficient solutions that can lower the overall operational costs. This is a great challenge because the one-time investment of installing a robotic system is extremely high compared to human operation. Efficient and optimal solutions that can lower the operational costs compared to human operation is thus vital.

We have presented several contributions that will increase the productivity and efficiency of the robotic operation of the platform. In Part III we presented a new approach that allows the manipulator to perform several different maintenance tasks such as spray painting, high-pressure water blasting, and welding in a lot shorter time and using less actuator torque. This part comes as a direct result of a desire from the oil companies who have recognised these tasks as the most important and time-consuming tasks expected to be performed by robots. We also presented solutions that allow the manipulator to take advantage of the motion of the base in the case when this is large enough to affect the manipulator dynamics. This will also save actuator torque and decrease the wear and tear on the robot.

We have also put considerable effort into verifying the theoretical results in the lab. The industry is only interested in solutions that work—not only in theory, but also in real life. We have thus presented several empirical studies from the lab that verify the theoretical studies. We have had access to a well equipped lab where we have performed experiments that illustrate, for the first time, how the inertial forces of a moving platform affect the manipulator. We have used this to show that the inertial forces acting on a robot mounted on a ship in only 1 meter wave height affect the robot dynamics and, if accounted for, can be used to improve the motion planner. We have also collected motion data from a real ship. This allows us to get valuable information about how well we can predict the future motion of a ship. This is important information when we use the predicted ship motion in the motion planner of robot. Finally we have performed several experiments that illustrate how the approach in Part III increases the efficiency of spray paint, welding, and high

pressure water blasting applications—not only in simulations, but also on an industrial robot in the lab. All these empirical studies strengthen the theoretical results presented throughout the thesis.

Future Work

As already noted, there is still a long way to go before off-shore oil platforms can be completely operated by robots. There are thus several interesting research topics that still need to be investigated. We believe the most challenging aspects of autonomous operation are related to robustness. Unfortunately, the oil fields that have not yet been explored are mostly in sensitive areas where the consequences of oil spill are enormous. This, combined with the fact that these areas are remotely located and often in extremely harsh environments, makes the robustness of the utilised systems a very interesting and a challenging research area that will require a lot of attention in the future. Robustness is also a major concern for the oil companies as unplanned shut-down of the platforms are extremely costly and should be avoided.

Also when it comes to the robotic manipulator we can expect several advances. Industrial robots are not built for extreme weather conditions. The robots will thus need to be manufactured for extreme conditions such as salt water environments, extreme temperatures, strong winds, explosive gases, and so on. New materials will probably have to be used, as well as improved coating to protect against low temperatures and salt water.

Finally we point out intelligent solutions as an important research area. Autonomous operation requires solutions that can take the right action also when unplanned or unforeseen events occur. Even though we cannot rely entirely on decisions made by the robotic systems, a large part of the decision-making process should be executed without the involvement of humans. This will require that the robotic system can make intelligent decisions on its own. The robotic system is also the eyes, ears, and nose of the on-shore human supervisor. The robotic system thus needs to make decisions on what information to pass on to the supervisor. Such intelligent decision-making also requires advances in areas such as machine learning and artificial intelligence.

One of the main goals of the TAIL IO project when it started four years ago was to study the feasibility of autonomously operated off-shore oil platforms. The time span was set to 15 years to build the first autonomous platform. Now, four years later, we believe that we can maintain this goal, i.e. that we will be able to build the first autonomous oil platform eleven years from now. A lot of research still remains, but the investment from both robotic manufacturers and oil companies, both when it comes to financial investment and the number of research projects on the area, suggests that this progress will continue and that relatively soon, the first fully autonomous off-shore oil platform will see the light of day.

Bibliography

Ahuactzin, J. M. and Gupka, K. K., 1999. The kinematic roadmap: A motion planning based global approach for inverse kinematics of redundant robots. *IEEE Transaction on Robotics and Automation*, 15(4), pp. 653–669.

Alpern, B., Carter, L., Grayson, M., and Pelkie, C., 1993. Orientation maps: Techniques for visualizing rotations (a consumers guide). *IEEE Conference on Visualization, San Jose, CA, USA*, pp. 183–188.

Angeles, J., 2004. The qualitative synthesis of parallel manipulators. *Journal of mechanical design*, 126(4), pp. 617–624.

Antonelli, G., 2006. *Underwater robots. Motion and force control of vehicle-manipulator systems*. Springer-Verlag, Berlin, Germany.

Antonio, J. K., 1994. Optimal trajectory planning for spray coating. *IEEE International Conference on Robotics and Automation, San Diego, CA, USA*, pp. 2570–2577.

Arai, H. and Tachi, S., 1990. Dynamic control of a manipulator with passive joints - position control experiments by a prototype manipulator. *IEEE/RSJ International Conference on Intelligent Robots and Systems, Ibaraki, Japan*, pp. 935–940.

Atkar, P. N., Greenfield, A., Conner, D. C., Choset, H., and Rizzi, A. A., 2005. Uniform coverage of automotive surface patches. *The International Journal of Robotics Research*, 24(11), pp. 883–898.

Balestrino, A., Maria, G. D., and Sciavicco, L., 1984. Robust control of robotic manipulators. *IFAC World Congress, Budapest, Hungary*, 5, pp. 2435–2440.

Bhat, S. P. and Bernstein, D. S., 2000. A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon. *Systems and Control Letters*, 39(1), pp. 63–70.

Bicchi, A. and Prattichizzo, D., 2000. Manipulability of cooperating robots with unactuated joints and closed-chain mechanisms. *IEEE Transaction of Robotics and Automation*, 16(4), pp. 336–345.

Bowling, A. and Khatib, O., 1997. Design of macro/mini manipulators for optimal dynamic performance. *IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, USA*, pp. 449–454.

Boyd, S., Ghaoui, L. E., Feron, E., and Balakrishnan, V., 1994. *Linear Matrix Inequalities in System and Control Theory*. SIAM Studies in Applied Mathematics, Philadelphia, PA, USA.

Boyd, S. and Vandenberghe, L., 2004. *Convex Optimization*. Cambridge University Press, New York, USA.

Boyd, S. P. and Wegbreit, B., 2007. Fast computation of optimal contact forces. *IEEE Transactions on Robotics*, 23(6), pp. 1117–1132.

Breivik, M., Hovland, G., and From, P. J., 2009. Trends in research and publication: Science 2.0 and open access. *Modeling, Identification and Control*, 30(3), pp. 181–190.

Børhaug, E., 2008. *Nonlinear Control and Synchronization of Mechanical Systems*. Ph.D. thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway.

Brockett, R. W., 1984. Robotic manipulators and the product of exponentials formula. *Mathematical Theory of Networks and systems*, pp. 120–129.

Bronshtein, I. N., Semendyayev, K. A., Musiol, G., and Muehlig, H., 2003. *Handbook of Mathematics*. Springer, Heidelberg, Germany.

Bullo, F. and Lewis, A. D., 2004. *Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control Systems*. Springer Verlag, New York, USA.

Buss, M., Hashimoto, H., and Moore, J. B., 1996. Dextrous hand grasping force optimization. *IEEE Transactions on robotics and automation*, 12(3), pp. 406–418.

Cameron, J. M. and Book, W. J., 1997. Modeling mechanisms with nonholonomic joints using the Boltzmann-Hamel equations. *The International Journal of Robotics Research*, 16(1), pp. 47–59.

Cannon, M., Couchman, P., and Kouvaritakis, B., 2007. MPC for stochastic systems. *Assessment and Future Directions of Nonlinear Model Predictive Control*, 358, pp. 255–268.

Chung, J. C., Bien, Z., and Kim, Y. S., 1990. A note on ship-motion prediction based on wave-excitation input estimation. *IEEE Journal of oceanic engineering*, 15(3), pp. 244–250.

Conner, D., Greenfield, A., Atkar, P., Rizzi, A., and Choset, H., 2005. Paint deposition modeling for trajectory planning on automotive surfaces. *Automation Science and Engineering*, 2(4), pp. 381–392.

Corke, P., 1996. A robotics toolbox for MATLAB. *IEEE Robotics and Automation Magazine*, 3(1), pp. 24–32.

Couchman, P., Cannon, M., and Kouvaritakis, B., 2006. Stochastic MPC with inequality stability constraints. *Automatica*, 42(12), pp. 2169–2174.

Craig, J. J., 1987. *Adaptive Control of Mechanical Manipulators*. Addison-Wesley Longman Publishing, Boston, MA.

Dai, J. S., Huang, Z., and Lipkin, H., 2006. Mobility of overconstrained parallel mechanisms. *Transactions of ASME*, 128(1), pp. 220–229.

D'Souza, A., Vijayakumar, S., and Schaal, S., 2001. Learning inverse kinematics. *IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii, USA*, pp. 298–303.

Dubowsky, S. and Papadopoulos, E., 1993. The kinematics, dynamics and control of free-flying and free-floating space robotic systems. *IEEE Transactions on Robotics and Automation*, 9(5), pp. 531–541.

Duindam, V., 2006. *Port-Based Modeling and Control for Efficient Bipedal Walking Robots*. Ph.D. thesis, University of Twente.

Duindam, V. and Stramigioli, S., 2007. Lagrangian dynamics of open multibody systems with generalized holonomic and nonholonomic joints. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA* pp. 3342–3347.

Duindam, V. and Stramigioli, S., 2008. Singularity-free dynamic equations of open-chain mechanisms with general holonomic and nonholonomic joints. *IEEE Transactions on Robotics*, 24(3), pp. 517–526.

Dwyer, T. A. W., 1984. Exact nonlinear control of large angle rotational maneuvers. *IEEE Transactions on Automatic Control*, 29(9), pp. 769–774.

Egeland, O. and Gravdahl, J. T., 2003. *Modeling and Simulation for Automatic Control*. Marine Cybernetics AS, Trondheim, Norway.

Egeland, O. and Pettersen, K. Y., 1998. Free-floating robotic systems. *Control Problems in Robotics and Automation*, 230, pp. 119–134.

Egeland, O. and Sagli, J. R., 1993. Coordination of motion in a spacecraft/manipulator system. *International Journal of Robotics Research*, 12(4), pp. 366–379.

English, J. D. and Maciejewski, A. A., 1998. Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures. *IEEE Transactions of Robotics and Automation*, 14(4), pp. 566–575.

Fossen, T. I., 1991. *Nonlinear Modelling and Control of Underwater Vehicles*. Ph.D. thesis, Department of Engineering Cybernetics, The Norwegian Institute of Technology.

Fossen, T. I., 2002. *Marine Control Systems, 3rd printing*. Marine Cybernetics AS, Trondheim, Norway.

Fossen, T. I., 2009. *Modeling and control of marine vessels*. To be published.

Fossen, T. I. and Fjellstad, O. E., 1995. Nonlinear modelling of marine vehicles in 6 degrees of freedom. *International Journal of Mathematical Modelling Systems*, 1(1), pp. 17–28.

From, P. J., 2006. *Modelling and Optimal Trajectory Planner for Industrial Spray Paint Robots*. Master's thesis, NTNU.

From, P. J., Duindam, V., Gravdahl, J. T., and Sastry, S., 2009a. Modeling and motion planning for mechanisms on a non-inertial base. *International Conference of Robotics and Automation, Kobe, Japan*, pp. 3320–3326.

From, P. J., Duindam, V., Pettersen, K. Y., Gravdahl, J. T., and Sastry, S., 2010a. Singularity-free dynamic equations of vehicle-manipulator systems. *Simulation Modelling Practice and Theory*, 18(6), pp. 712–731.

From, P. J. and Gravdahl, J. T., 2007a. General solutions to kinematic and functional redundancy. *IEEE Conference on Decision and Control, New Orleans, LA, USA*, pp. 5779–5786.

From, P. J. and Gravdahl, J. T., 2007b. Representing attitudes as sets of frames. *American Control Conference, New York, USA*, pp. 2465–2472.

From, P. J. and Gravdahl, J. T., 2008a. Considering passive joints as functional redundancy. *IFAC world congress, Seoul, South Korea*.

From, P. J. and Gravdahl, J. T., 2008b. General solutions to kinematic and functional redundancy. *Modeling, Identification and Control*, 29(2), pp. 39–50.

From, P. J. and Gravdahl, J. T., 2008c. Iterative solutions to the inverse geometric problem for manipulators with no closed form solution. *Modeling, Identification and Control*, 29(3), pp. 77–92.

From, P. J. and Gravdahl, J. T., 2008d. On the equivalence of orientation error and positive definiteness of matrices. *International Conference on Control, Automation, Robotics and Vision, Hanoi, Vietnam*, pp. 2089–2094.

From, P. J. and Gravdahl, J. T., 2008e. On the mobility and fault tolerance of closed chain manipulators with passive joints. *Modeling, Identification and Control*, 29(4), pp. 151–165.

From, P. J. and Gravdahl, J. T., 2008f. Real-time optimal trajectory planning for robotic manipulators with functional redundancy. *IEEE International Conference on Industrial Technology, Gippsland, VIC , Australia*, pp. 1–7.

From, P. J. and Gravdahl, J. T., 2009a. Fault tolerance of parallel manipulators with passive joints. *IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Barcelona, Spain*.

From, P. J. and Gravdahl, J. T., 2009b. A geometric approach to handling torque failure in serial manipulators and closed chain manipulators with passive joints. *Submitted to ASME Journal of Mechanical Design*.

From, P. J. and Gravdahl, J. T., 2009c. Representing sets of orientations as convex cones. *IEEE International Conference on Robotics and Automation, Kobe, Japan*, pp. 1352–1357.

From, P. J. and Gravdahl, J. T., 2010a. Optimal paint gun orientation in spray paint applications - experimental results. *Submitted to IEEE Transactions of Automation Science and Engineering*.

From, P. J. and Gravdahl, J. T., 2010b. A real-time algorithm to determine the optimal paint gun orientation in spray paint applications. *Accepted for publication in IEEE Transactions on Automation Science and Engineering*.

From, P. J., Gravdahl, J. T., and Abbeel, P., 2009b. Motion planning and control of robotic manipulators on seaborne platforms. *Submitted to Control Engineering Practice*.

From, P. J., Gravdahl, J. T., and Abbeel, P., 2010b. On the influence of ship motion prediction accuracy on motion planning and control of robotic manipulators on seaborne platforms. *International Conference of Robotics and Automation, Anchorage, Alaska, USA*.

From, P. J., Pettersen, K. Y., and Gravdahl, J. T., 2010c. Singularity-free dynamic equations of spacecraft-manipulator systems. *IFAC Symposium on Automatic Control in Aerospace, Prague, Czech Republic*.

From, P. J., Pettersen, K. Y., and Gravdahl, J. T., 2010d. Singularity-free dynamic equations of AUV-manipulator systems. *Symposium on Intelligent Autonomous Vehicles, Lecce, Italia*.

From, P. J., Pettersen, K. Y., and Gravdahl, J. T., 2010e. Singularity-free formulation of the dynamically equivalent manipulator mapping for free-flying and free-floating space manipulators. *To be submitted to AAIA Journal of Guidance, Control, and Dynamics*.

From, P. J., Pettersen, K. Y., and Gravdahl, J. T., 2011. Singularity-free formulation of the dynamically equivalent manipulator mapping for space manipulators. *To be submitted*.

From, P. J., Schjølberg, I., Gravdahl, J. T., Pettersen, K. Y., and Fossen, T. I., 2010f. On the boundedness and skew-symmetric properties of the inertia and Coriolis matrices for vehicle-manipulator systems. *Submitted to Systems and Control Letters*.

From, P. J., Schjølberg, I., Gravdahl, J. T., Pettersen, K. Y., and Fossen, T. I., 2010g. On the boundedness property of the inertia matrix and skew-symmetric property of the Coriolis matrix for vehicle-manipulator systems. *Symposium on Intelligent Autonomous Vehicles*.

Ghorbel, F., Srinivasan, B., and Spong, M. W., 1998. On the uniform boundedness of the inertia matrix of serial robot manipulators. *Journal of Robotic Systems*, 15(1), pp. 17–28.

Gibbs, J. W., 1879. On the fundamental formulae of dynamics. *American Journal of Mathematics*, 2(1), pp. 49–64.

Gingsberg, J., 2007. *Engineering dynamics*. Cambridge University Press, New York, USA.

Goldstein, H., Poole, C. P., and Safko, J. L., 2001. *Classical Mechanics*. Addison Wesley, San Francisco, USA.

Grudic, G. Z. and Lawrence, P. D., 1993. Iterative inverse kinematics with manipulator configuration. *IEEE Transactions on Robotics and Automation*, 9(4), pp. 476–483.

van der Ha, J. C. and Shuster, M., 2009. A tutorial on vectors and attitude. *IEEE Control Systems Magazine*, 29(2), pp. 94–107.

Han, L., Trinkle, J., and Li, Z., 2000. Grasp analysis as linear matrix inequalitiy problem. *IEEE Transactions on Robotics and Automation*, 16(6), pp. 663–674.

Hanson, A. J., 2006. *Visualizing Quaternions*. Morgan Kaufmann, San Francisco, USA.

Harrison, C. A. and Qin, S. J., 2009. Minimum variance performance map for constrained model predictive control. *Journal of Process Control*, 19, pp. 1199–1204.

Hauser, J. and Murray, R. M., 1990. Nonlinear controllers for non-integrable systems: the acrobot example. *American Control Conference, San Diego, CA, USA*, pp. 669–671.

Herman, P. and Kozlowski, K., 2006. A survey of equations of motion in terms of inertial quasi-velocities for serial manipulators. *Archive of Applied Mechanics*, 76(9-10), pp. 579–614.

Hertling, P., Hog, L., Larsen, R., Perram, J., and Petersen, H. G., 1996. Task curve planning for painting robots - part 1: process modeling and calibration. *IEEE Transactions on Robotics and Automation*, 12(2), pp. 157–352.

Hervé, J. M., 1978. Analyse structurelle des mécanismes par groupe des déplacements. *Mechanism Machine Theory*, 13(4), pp. 435–437.

Hervé, J. M. and Sparacino, F., 1991. Structural synthesis of parallel robots generating spatial translation. *IEEE International Conference on Advances in Robotics, Pisa, Italy*, pp. 808–813.

Hughes, P. C., 2002. *Spacecraft Attitude Dynamics*. Dover Publications, Mineola, NY, USA.

Jarzbowska, E., 2008. Quasi-coordinates based dynamics modeling and control design for nonholonomic systems. *Nonlinear Analysis*, 71(12), pp. 118–131.

Johnson, M. P., 1995. *Exploiting Quaternions to Support Expressive Interactive Character Motion*. Ph.D. thesis, Massachusetts Institute of Technology.

Kahan, W., 1983. Lectures on computational aspects of geometry. *Department of electrical engineering and computer sciences, University of California, Berekeley. Unpublished*.

Kajita, H. and Kosuge, K., 1997. Force control of robot floating on the water utilizing vehicle restoring force. *IEEE/RSJ International Conference on Intelligent Robot and Systems, Grenoble, France*, 1, pp. 162–167.

Kane, T. R. and Levinson, D. A., 1985. *Dynamics: theory and applications*. McGraw Hill, New York, USA.

Kane, T. R., Likins, P. W., and Levinson, D. A., 1983. *Spacecraft dynamics*. McGraw Hill, New York, USA.

Khalil, W. and Dombre, E., 2002. *Modeling, Identification and Control of Robots*. Hermes Penton, London, UK.

Khan, A., Bil, C., and Marion, K. E., 2005. Ship motion prediction for launch and recovery of air vehicles. *MTS/IEEE OCEANS, Washington CD, USA*, pp. 2795–2801.

Kim, T. and Sarma, S. E., 2003. Optimal sweeping paths on a 2-manifold: A new class of optimization problems defined by path structures. *IEEE Transactions on Robotics and Automation*, 19(4), pp. 613–636.

Kitarovic, J., Tomas, V., and Cisic, D., 2005. The electronic and informatics age - a new stage in developing highly effective ships. *International ELMAR Symposium, Zadar, Croatia*, pp. 385–388.

Kosuge, K., Okuda, M., and Fukuda, T., 1992. Motion control of manipulator/vehicle system floating on water. *IEEE International Workshop on Advanced Motion Control, Nagoya, Japan*, pp. 506–511.

Kozlowski, K. and Herman, P., 2008. Control of robot manipulators in terms of quasi-velocities. *Journal of Intelligent and Robotic Systems*, 53(3), pp. 205–221.

Kristiansen, R., Nicklasson, P. J., and Gravdahl, J. T., 2005. Satellite attitude control by quaternion-based backstepping. *IEEE Transactions on control systems technology*, 17(1), pp. 227–232.

Kuipers, J. B., 2002. *Quaternions and Rotation Sequences*. Princeton University Press, Princeton, New Jersey, USA.

Kwatny, H. G. and Blankenship, G., 2000. *Nonlinear Control and Analytical Mechanics A Computational Approach*. Birkhäuser, Boston, USA.

Lebans, G., Wilkie, K., Dubay, R., Crabtree, D., and Edmonds, T., 1997. Telerobotic shipboard handling system. *MTS/IEEE OCEANS, Halifax, Nova Scotia, Canada*, pp. 1237–1241.

Lee, C. C. and Hervé, J. M., 2006. Translational parallel manipulators with doubly planar limbs. *Mechanism and Machine Theory*, 41(4), pp. 359–486.

Lesser, M., 1992. A geometrical interpretation of kanes equations. *Mathematical and Physical Sciences*, 436(1896), pp. 69–87.

Lewis, A. D., 1996. The geometry of the Gibbs-Appel equations and Gauss's principle of least constraint. *Reports on Mathematical Physics*, 38(1), pp. 11–28.

Li, Q. C., Huang, Z., and Hervé, J. M., 2004. Type synthesis of 3R2T 5-DoF parallel mechanism using the Lie group of displacement. *IEEE Transactions on Robotics and Automation*, 20(2), pp. 181–190.

Liang, B., Xu, Y., and Bergerman, M., 1998. Mapping a space manipulator to a dynamically equivalent manipulator. *ASME Journal of Dynamic Systems, Measurement, and Control*, 120(1), pp. 1–7.

Liang, B., Xu, Y., Bergerman, M., and Li, G., 1997. Dynamically equivalent manipulator for space manipulator system: Part 2. *EEE/RSJ International Conference on Intelligent Robots and Systems, Grenoble, France*.

Lin, Q. and Burdick, J. W., 2000. Objective and frame-invariant kinematic metric functions for rigid bodies. *International Journal of Robotics Research*, 19(6), pp. 612–625.

Lipkin, H. and Duffy, J., 2002. Sir Robert Stawell Ball and methodologies of modern screw theory. *Journal of Mechanical Engineering Science*, 216(1), pp. 1–11.

Liu, Y. H., Xu, Y., and Bergerman, M., 1999. Cooperation control of multiple manipulators with passive joints. *IEEE Transactions on Robotics and Automation*, 15(2), pp. 258–267.

Love, L. J., Jansen, J. F., and Pin, F. G., 2004. On the modeling of robots operating on ships. *IEEE International Conference on Robotics and Automation, New Orleans, LA, USA*, pp. 2436–2443.

Luenberger, D. G., 2003. *Linear and Nonlinear Programming*. Kluwer Academic Publishers, Dordrecht, Nederland.

Luh, J., Walker, M., and Paul, R., 1980. On-line computational scheme for mechanical manipulators. *ASME Journal of Dynamical Systems, Measurement, and Control*, 102(2), pp. 69–76.

Marsden, J. E. and Ratiu, T. S., 1999. *Introduction to Mechanics and Symmetry, 2nd ed., Texts in Applied Mathematics*. Springer Verlag, New York, USA.

Maruskin, J. M. and Bloch, A. M., 2007. The Boltzmann-Hamel equations for optimal control. *IEEE Conference on Decision and Control, San Diego, CA, USA*, pp. 554–559.

Matone, R. and Roth, B., 1999. In-parallel manipulators: A framework on how to model actuation schemes and a study of their effects on singular postures. *Transaction of ASME*, 121(1), pp. 2–8.

McLachlan, R. I. and Quispel, G. R. W., 2006. Geometric integrators for ODEs. *Journal of Physics A: Mathematical and General*, 39(19), pp. 5251–5285.

McMillan, S., Orin, D. E., and McGhee, R. B., 1995. Efficient dynamic simulation of an underwater vehicle with a robotic manipulator. *IEEE Transactions on systems, man and cybernetics*, 25(8), pp. 1194–1206.

Meirovich, L. and Kwak, M. K., 1989. State equations for a spacecraft with maneuvering flexible appendages in terms of quasi-coordinates. *Applied Mechanics Reviews*, 42(11), pp. 161–170.

Meng, J., Liu, G., and Li, Z., 2007. A geometric theory for analysis and synthesis of sub-6 DoF parallel manipulators. *IEEE Transactions on robotics*, 23(4), pp. 625–649.

Moosavian, S. A. A. and Papadopoulos, E., 2004. Explicit dynamics of space free-flyers with multiple manipulators via spacemaple. *Advanced robotics*, 18(2), pp. 223–244.

Moosavian, S. A. A. and Papadopoulos, E., 2007. Free-flying robots in space: an overview of dynamics modeling, planning and control. *Robotica*, 25, pp. 537–547.

Muenchhof, M., Becka, M., and Isermanna, R., 2009. Fault-tolerant actuators and drives-structures, fault detection principles and applications. *Annual Reviews in Control*, 33(2), pp. 136–148.

Munthe-Kaas, H., 1998. Runge-kutta methods on lie groups. *BIT Numerical Mathematics*, 38(1), pp. 92–111.

Murray, R. M., Li, Z., and Sastry, S. S., 1994. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, FL, USA.

Nakamura, Y. and Hanafusa, H., 1986. Inverse kinematics solutions with singularity robustness for robot manipulator control. *ASME Journal of Dynamic Systems, Measurement, and Control*, 108, pp. 163–171.

Oh, S.-R., Mankala, K., Agrawal, S., and Albus, J., 2005. Dynamic modeling and robust controller design of a two-stage parallel cable robot. *Multibody System Dynamics*, 13(4), pp. 385–399.

Oriolo, G. and Nakamura, Y., 1991. Free-joint manipulators: Motion control under second-order nonholonomic constraints. *IEEE International Workshop on Intelligent Robots and Systems, Osaka, Japan*, pp. 1248–1253.

Oyama, E., Chong, N. Y., Agah, A., Maeda, T., and Tachi, S., 2001. Inverse kinematics learning by modular architecture neural networks with performance prediction networks. *IEEE International Conference on Robotics and Automation, Seoul, Korea*, pp. 1006–1012.

Paden, B. and Sastry, S. S., 1988. Optimal kinematic design of 6R manipulators. *International Journal of Robotics Research*, 7(2), pp. 43–61.

Papadopoulos, E. and Dubowsky, S., 1991. On the nature of control algorithms for free-floating space manipulators. *IEEE Transaction Robotics and Automation*, 7(6), pp. 750–758.

Park, F. C., Bobrow, J. E., and Ploen, S. R., 1995. A Lie group formulation of robot dynamics. *International Journal of Robotics Research*, 14(6), pp. 609–618.

Parlaktuna, O. and Ozkan, M., 2004. Adaptive control of free-floating space manipulators using dynamically equivalent manipulator model. *Robotics and Autonomous Systems*, 46(3), pp. 185–183.

Perdereau, V., Passi, C., and Drouin, M., 2002. Real-time control of redundant robotic manipulators for mobile obstacle avoidance. *Robotics and Autonomous Systems*, 41(1), pp. 41–59.

Potkonjak, V., Dordevic, G., Kostic, D., and Rasic, M., 2000. Dynamics of anthropomorphic painting robot: Quality analysis and cost reduction. *Robotics and Autonomous Systems*, 32(1), pp. 1493–1499.

Ramabhadran, R. and Antonio, J. K., 1997. Fast solution techniques for a class of optimal trajectory planning problems with applications to automated spray coating. *IEEE Transactions of Robotics and Automation*, 13(4), pp. 519–530.

Reuleaux, F., 1875. *Theoretische Kinematik*. Dover, New York, USA.

Rico, J. M., Aguilera, L. D., Gallardo, J., Rodriguez, R., Orozco, H., and Barrera, J. M., 2006. A more general mobility criterion for parallel mechanisms. *Journal of Mechanical Design*, 128(1), pp. 207–219.

Rico, J. M., Gallardo, J., and Ravani, B., 2003. Lie algebra and the mobility of kinematic chains. *Journal of Robotic Systems*, 20(8), pp. 477–499.

Rossmann, W., 2002. *Lie Groups - An introduction through linear algebra*. Oxford science publications, Oxford, UK.

Sagatun, S. I. and Fossen, T. I., 1992. Lagrangian formulation of underwater vehicles. *Conference of Systems, Man and Cybernetics, Charlottesville, VA, USA*, pp. 1029–1034.

Salvesen, N., Tuck, E. O., and Faltinsen, O. M., 1970. Ship motions and sea loads. *Transaction of SNAME*, 78, pp. 250–287.

Salzmann, D. C., 2007. Ampelmann prototype - developing a motion compensating platform for offshore access. *European Wind Energy Conference, Milano, Italy*.

Schjølberg, I., 1996. *Modeling and control of underwater robotic systems*. Ph.D. thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway.

Schjølberg, I. and Fossen, T. I., 1994. Modelling and control of underwater vehicle-manipulator systems. *Conference on Marine Craft Maneuvering and Control, Southampton, UK*, pp. 45–57.

Sciavicco, L. and Siciliano, B., 2005. *Modelling and Control of Robot Manipulators*. Springer Verlag, New York, USA.

Selig, J. M., 2000. *Geometric fundamentals of robotics*. Springer Verlag, New York, USA.

Skourup, C. and Pretlove, J., 2009. The robotized field operator: Greater safety and productivity by design. *ABB Review*, 1, pp. 68–73.

Smith, T. S., Farouki, R. T., al Kandari, M., and Pottmann, H., 2001. Optimal slicing of free-form surfaces. *Computer Aided Geometric Design*, 19(1), pp. 43–64.

Soanes, C. and Stevenson, A., editors, 2008. *Concise Oxford English Dictionary*. Oxford University Press.

Steffen, T., Michail, K., Dixon, R., Zolotas, A. C., and Goodall, R. M., 2009. Optimal passive fault tolerant control of a high redundancy actuator. *IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Barcelona, Spain*.

Åstrøm, K. J., 1967. Computer control of a paper machine - an application of linear stochastic control theory. *IBM Journal of Research and Development*, 7, pp. 389–396.

Suh, S.-H. and I.-K. Woo, S.-K. N., 1991. Development of an automatic trajectory planning system (ATPS) for spray painting robots. *IEEE International Conference on Robotics and Automation, Sacramento, California*, pp. 1948–1955.

Tanner, H. G. and Kyriakopoulos, K. J., 2001. Mobile manipulator modeling with Kane's approach. *Robotica*, 19, pp. 675–690.

Tevatia, G. and Schaal, S., 2000. Inverse kinematics for humanoid robots. *IEEE International Conference on Robotics and Automation, San Francisco, CA, USA*, pp. 294–299.

Tinós, R. and Terra, M. H., 2002. Control of cooperative manipulators with passive joints. *American Control Conference, Anchorage, Alaska, USA*, pp. 1129–1134.

Tinós, R., Terra, M. H., and Bergerman, M., 2007. A fault tolerance framework for cooperative robotic manipulators. *Control Engineering Practice*, 15, pp. 615–625.

Vafa, Z. and Dubowsky, S., 1987. On the dynamics of manipulators in space using the virtual manipulator approach. *IEEE International Conference on Robotics and Automation, North Carolina, USA*, pp. 579–585.

Vandenberghe, L., Boyd, S., and Wu, S.-P., 1996. Determinant maximization with linear matrix inequality constraints. *http://stanford.edu/∼boyd/papers/maxdet.html, download date: September 1. 2008*.

Vatland, S. and Svenes, M., 2008. Tail integrated operations, project newsletter. *http://www-05.ibm.com/no/solutions/chemicalspetroleum/pdf/tailnewsletter_feb2008.pdf, download date: July 1. 2009*.

Wang, L.-C. T. and Chen, C. C., 1991. A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 7(4), pp. 489–499.

Welman, C., 1993. *Inverse kinematics and geometric constraints for articulated figure manipulation*. Master's thesis, Simon Frasier University, Burnaby, BC, Canada.

Wen, J. T.-Y. and Kreutz-Delgado, K., 1991. The attitude control problem. *IEEE Transactions on Automatic Control*, 30(10), pp. 1148–1162.

Whitney, D. E., 1969. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, 10(2), pp. 47–53.

de Wit, C. C., Diaz, E., and Perrier, M., 1998. Robust nonlinear control of an underwater vehicle/manipulator system with composite dynamics. *IEEE International conference on robotics and automation, Leuven, Belgium*, pp. 452–457.

Xian, B., de Queiroz, M., Dawson, D., and Walker, I., 2004. Task-space tracking control of robot manipulators via quaternion feedback. *IEEE Transactions on Robotics and Automation*, 20(1), pp. 160–167.

Yang, X., Pota, H., Garratt, M., and Ugrinovskii, V., 2008a. Prediction of vertical motions for landing operations of uavs. *IEEE Conference on Decision and Control, Cancun, Mexico*, pp. 5058–5063.

Yang, X., Pota, H., Garratt, M., and Ugrinovskii, V., 2008b. Ship motion prediction for maritime flight operations. *IFAC World Congress, Seoul, Korea*, pp. 12407–12412.

Yoshikawa, T., Harada, K., and Matsumoto, A., 1996. Hybrid position/force control of flexible-macro/rigid-micro manipulator systems. *IEEE Transactions on Robotics and Automation*, 12(4), pp. 633–640.

Yuan, J. S. C., 1988. Closed-loop manipulator control using quaternion feedback. *IEEE Journal of Robotics Automation*, 4(4), pp. 434–440.

Zhang, Q. and Li, S., 2007. Enhanced performance assessment of subspace model-based predictive controller with parameters tuning. *The Canadian journal of chemical engineering*, 85, pp. 537–548.

Zhao, J. and Badler, N. I., 1994. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics*, 13, pp. 313–336.