NTNU

Norwegian University of
Science and Technology

Hybrid Force Motion Synchronization Control of Robot Manipulators

Kristoffer Fikkan

Master of Science in Engineering Cybernetics
Submission date: June 2010
Supervisor: Kristin Ytterstad Pettersen, ITK
Co-supervisor: Erik Kyrkjebø, SINTEF Anvendt kybernetikk

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

Synchronization through coordinated and cooperative control is an active topic of research, and of high industrial interest as it allows for robotic solutions in production and manufacturing providing increased speed, cost-efficiency and safety. Furthermore, it is a an enabling technology for unmanned oil and gas platforms that is of high interest to the oil and gas industry in Norway as it is moving towards smaller unmanned platforms in order to utilize the remaining oil and gas resources on the Norwegian continental shelf. In this project you will acquaint yourself with coordinated and cooperative control, in particular for robot manipulators, and develop control systems for robot manipulators that will enable robot manipulators to cooperate and manipulate dynamic objects in the environment. Synchronization control is related to tracking control in that the robot must move in time with the reference, but differs in that the reference in synchronization control is the motion of the physical manipulator rather than an ideal precomputed trajectory. There is already some work on synchronization of robot manipulators in [1] and [2], but none that incorporates the synchronization concept into contact operations for controlling force and motion at the same time.

The main goal of the project is to utilize ideas from synchronization theory together with the Hybrid Force/Motion Control concept as introduced in [3] to separate the interaction control problem into two separate subproblems; motion control and force control of the contact forces applied to the environment. Based on the results from the pre-study project, where a control concept was proposed for linking motion control to interaction with the environment in terms of synchronized behaviors.

1. Perform a rigorous analysis of the control concept in terms of stability.
(In particular, note that with the proposed control concept, only the position/velocity error in the directions without contact are seeked control, together with the force in the directions with contact. It remains to analyze the internal dynamics Sfx, i.e. the behavior of the error states in the directions in which there is contact. Can we prove that these are bounded? Also consider the global stability properties.)
2. Verify the analysis by simulations.
3. Discuss the control concept and its consequences (Including stability properties, implications for practical use and implementation issues)
4. Perform a literature study to establish state-of-the art for force control together with stability analysis (Lyapunov, linear and other analysis tools). Compare the proposed control concept, its stability properties and practical consequences to existing force control solutions.

---

[1] H. Nijmeijer, Rodriguez-Angeles, A., Synchronization of Mechanical Systems, vol. 46: World Scientific Series on Nonlinear Science, Series A, 2003.
[2] E. Kyrkjebø, "Motion Coordination of Mechanical Systems: Leader-Follower Synchronization Control of Euler-Lagrange Systems using Output Feedback Control," Norwegian University of Science and Technology, 2007.
[3] M. H. Raibert, Craig, J. J., "Hybrid position/force control of manipulators," ASME Journal of Dynamic Systems, Measurement and Control, vol. 103, pp. 126 - 133, 1981.

Assignment given: 11. January 2010
Supervisor: Kristin Ytterstad Pettersen, ITK

NTNU
Department of Engineering Cybernetics

Hybrid Force Motion Synchronization Control of Robot Manipulators

Kristoffer Fikkan

Norwegian University of Science and Technology
Department of Engineering Cybernetics

June 2010

Supervisor
Kristin Y. Pettersen

Co-Supervisor
Erik Kyrkjebø

Preface

This master's thesis presents the work done during the spring of 2010 as a part of my Master of Science degree at the department of Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU). This thesis is a continuation of the specialization project I carried out during the fall of 2009.

I want to thank my supervisor, Professor Kristin Y. Pettersen, for her inspiration and support. Her knowledge has been invaluable.

I also want to thank my co-supervisor, PhD Erik Kyrkjebø, for his interesting angle of approach to problems. His contribution is sincerely appreciated.

Thank you for your confidence in me.

_____

Kristoffer Fikkan

Trondheim, June 14, 2010

Summary

The main objective of this master's thesis was to combine the theory on synchronization of robot manipulators with the concept of hybrid force/motion control; resulting in a controller capable of following both the trajectory of another robot and a desired force trajectory at the same time.

This report includes a short introduction to synchronization theory for robot manipulators, and a more thorough summary of existing hybrid control schemes.

An intuitive method for describing constraints caused by the environment is presented, and this leads to a straightforward way to separate the interaction control problem into two subproblems. When the manipulator end-effector is in contact with a surface, the hybrid controller tracks a position reference along the surface and a force reference normal to the surface.

The contribution of this master's thesis is a proposed hybrid controller, incorporating ideas from previous hybrid schemes. The closed loop system is shown to be globally asymptotically stable in the position controlled degrees of freedom, and bounded in the force controlled degrees of freedom. The performance of the proposed hybrid control concept is demonstrated by simulations.

The proposed hybrid controller easily reduces to a pure position controller when desired.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Synchronization control of robot manipulators is beneficial for the industry as it allows more than one manipulator to do complex tasks together. For example, assembling of products or lifting of heavy objects can be done with higher efficiency and precision. In addition, synchronization control is a prerequisite for smaller unmanned oil and gas platforms, which is needed to utilize the remaining oil and gas on the Norwegian continental shelf.

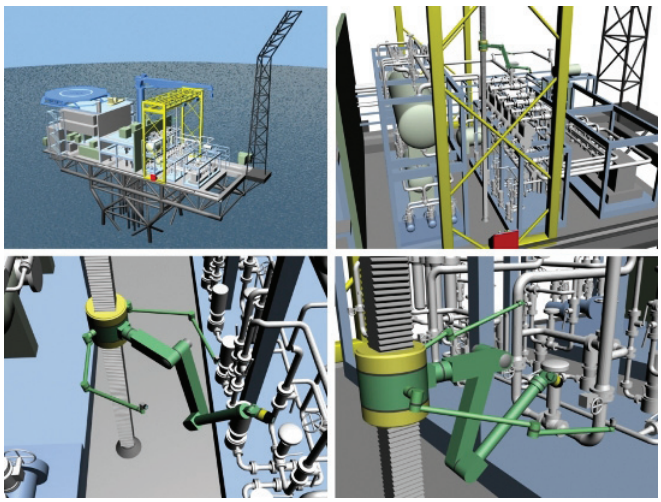

Figure 1.1: Illustration of the unmanned platform concept. Picture from `www.sintef.no`.

The synchronization concept is somewhat similar to the task of tracking a reference in time. However, instead of following a desired precomputed trajectory, the reference is the motion of a physical manipulator. Synchronization control can be implemented as coordinated or cooperative control. In a *coordinated control* scheme one of the manipulators is defined as leader while the other ones are followers. The leader tracks a predefined trajectory while the followers focus on moving in sync with the leader. When the manipulators are working *cooperatively*, they all work together to accomplish both the tracking goal and the synchronization goal.

Pure position control is sufficient for some tracking tasks. For example, in spray painting and spot welding the robot manipulator is not directly exerting forces on objects in the environment. On the other hand, when physical contact is needed, as in assembling or polishing, control of the forces between the robot manipulator end effector and the surface are advantageous. In this way a certain force against the surface may be sustained and damage to the end effector or the environment is avoided. In a pure position control system small deviations from the desired trajectory may cause loss of contact with the surface or severely large forces at the end effector.

Hybrid force/motion control allows a robot manipulator to follow a position trajectory and simultaneously adjust the forces applied to the environment based on measurements from force sensors, treating them as two separate subproblems.

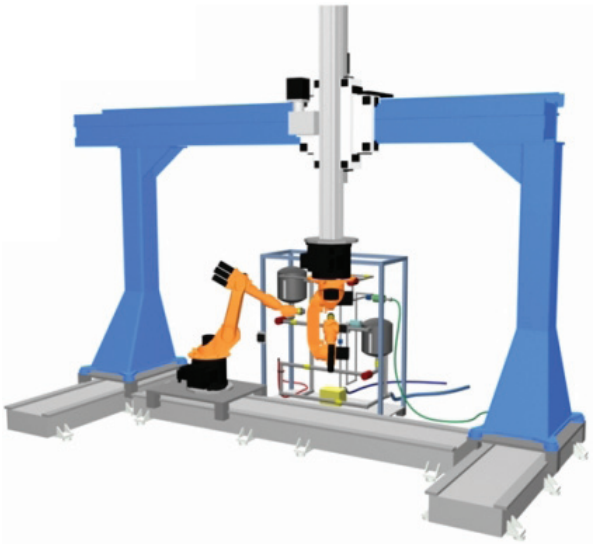

Figure 1.2: The SINTEF robotic lab facility. Picture from `www.sintef.no`.

1.2 The goal of this thesis

This master's thesis is a continuation of my pre-study project, where ideas from synchronization theory were utilized together with the concept of hybrid force/motion control. The interaction control problem was separated into two subproblems: position/velocity control along a constraint surface and force control normal to this surface.

The project resulted in a hybrid controller for a robot manipulator in a coordinated synchronization scheme. By transforming measurements and reference trajectories to a conveniently placed constraint frame, the motion controlled and force controlled degrees of freedom were easily separated.

However, the choice of orientation representation was not addressed. This will affect the transformation of measurements from joint space, and may introduce singularities. This concern will be discussed in this thesis, and an updated hybrid controller will be proposed. The closed loop stability properties of the control concept will be investigated.

With the proposed control concept, only the position/velocity errors in the directions without contact, and the force errors in the directions with contact, are controlled. This ensures that the position/velocity errors in the normal direction do not appear as an disturbance for the controller. On the other hand, the dynamics of the position error states not controlled must be analyzed; hopefully they can be shown to be bounded.

To verify the analysis of the proposed control concept, MATLAB/Simulink will be used to perform simulations. I will implement a 6-degree-of-freedom robot manipulator with its kinematics and dynamics, and simulate various tasks where the robot is to follow a time-varying position reference and at the same time apply a force to the environment. A discussion of the performance of the hybrid controller will follow.

The proposed hybrid control concept will be discussed regarding the stability properties and its consequences for implementation and practical use. A literature study will be performed on existing force control concepts, and they will be compared to the proposed hybrid control concept.

1.3 Outline of this report

The following chapter begins with a short introduction to the Euler-Lagrange dynamic model for robot manipulators, followed by a brief summary of the concept of synchronization control.

Then follows a chapter dedicated to force control. The distinction between *impedance control* and *hybrid control* will be made clear, and two different established approaches to the hybrid problem will be presented.

Before the derivation of the proposed hybrid controller, the different choices and assumptions made will be stated. Subjects such as whether to control in joint space or task space, and how to represent the orientation of the manipulator end-effector will be addressed. Then the derivation of the hybrid controller itself is presented, followed by a stability analysis. This chapter ends with a comparison between the different control concepts mentioned in this master's thesis.

After the derivation of the proposed hybrid controller, the analysis will be verified by simulations. The results will be discussed in light of the theoretical analysis, and issues regarding practical use will be commented.

Chapter 2

Preliminaries

2.1 Modeling of robot manipulators

This section will give a brief introduction to the Euler-Lagrange equations of motion for robot manipulators. For details on the derivation, see Spong et al. [10]. The model is expressed in joint angles $\boldsymbol{q}$ and their derivatives $\dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$.

2.1.1 Dynamic model

The dynamics of a robot manipulator with n joint is given as

$$\mathbf{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{2.1}$$

where $\boldsymbol{q} \in \mathbb{R}^n$ is the joint coordinate vector, $\mathbf{M} \in \mathbb{R}^{n \times n}$ is the configuration dependent inertia matrix which is symmetric and positive definite for all $\boldsymbol{q} \in \mathbb{R}^n$, $\mathbf{C} \in \mathbb{R}^{n \times n}$ is the centrifugal and Coriolis matrix, $\boldsymbol{g} \in \mathbb{R}^n$ is the gravity force vector and $\boldsymbol{\tau} \in \mathbb{R}^n$ is the joint torque vector. This torque vector will be used to control the manipulator to a desired position and velocity.

The model (2.1) is based on the assumption of no friction in the joints. By assuming that the friction effects are non-neglectable but known, they can be ideally compensated for by the controller, resulting in the model represented by (2.1).

If the friction effects are not known, but are modeled, the manipulator model are modified as

$$\mathbf{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}(\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{2.2}$$

where $\boldsymbol{D} \in \mathbb{R}^n$ is a vector representing the sum of the friction effects, and are a function only dependent on the joint velocities $\dot{\boldsymbol{q}}$. The focus of this thesis is not to model friction; thus, for all purposes it is assumed that the friction effects are known and fully compensated for. For more on friction effects and how to model them, see [3].

2.1.2 External force at end-effector

The dynamic model given by (2.1) is adequate when the robot manipulator moves through free space without interacting directly with the environment. But as soon as the end effector touches an object, forces will be exerted on the robot manipulator.

The torque acting on the different joints of the robot resulting from this force is expressed by the transpose of the geometric Jacobian [8] [10]; this can be derived from the principle of virtual work [10]. By modifying Equation (2.1) we obtain

$$\mathbf{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \mathbf{J}^{\mathrm{T}}(\boldsymbol{q})\boldsymbol{F} = \boldsymbol{\tau} \tag{2.3}$$

where $\mathbf{J} \in \mathbb{R}^{6\times n}$ is the geometric Jacobian of the manipulator and $\boldsymbol{F} \in \mathbb{R}^6$ is the force[1] the end-effector is applying to the environment. This term appears only when the manipulator is in contact with the environment. Otherwise, the model is reduced to (2.1).

[1] In this thesis I will use 'position' and 'force' in the meaning of 'position and orientation' and 'force and moment', respectively.

2.2 Synchronization control

This section is a short introduction to synchronization control. Since there already are some work on synchronization of robot manipulators, for example [5] and [7], I chose to briefly mention the concept, and focus more on combining synchronization theory with force control.

The concept of synchronization control can be divided into *coordinated control* and *cooperative control*.

2.2.1 Coordinated control

Coordinated control is also known as a leader-follower scheme [5], or external synchronization [7]. A multi-robot system consisting of two or more fully actuated robot manipulators with n rigid joints is considered.

One of the robots is defined as the *leader*, and its motion is independent of the other robots. The other robots are labeled *followers*, and is responsible for the coordination of the system [5]. The leader robot is driven by a torque $\boldsymbol{\tau}_l \in \mathbb{R}^n$ to assure convergence of the joint position and velocities $\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l \in \mathbb{R}^n$ to a desired trajectory $\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d$. This input torque $\boldsymbol{\tau}_l$ and the dynamic model of the leader is unknown to the followers. The concept is illustrated in Figure 2.1.

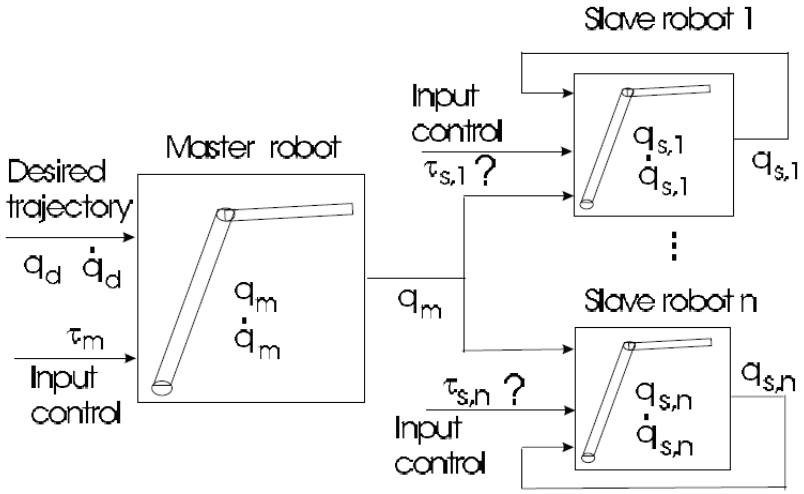

Figure 2.1: Coordinated control. Figure from [7].

The goal of a coordinative control scheme is to design a control law $\boldsymbol{\tau}_f$ for the follower, such that its joint position and velocities $\boldsymbol{q}_f, \dot{\boldsymbol{q}}_f$ synchronize with the leader's $\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l$. This control law $\boldsymbol{\tau}_f$ can only depend on the available measurements, which is assumed to be the position, velocity and acceleration measurements of both robots. Creating an estimator for unmeasured states is outside the scope of this thesis.

Each follower use the actual trajectory of the leader, $\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l$, as their desired trajectory, instead of using the leader's desired trajectory $\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d$ as reference. This ensures that the followers synchronize with the leader independently of whether the leader has converged to the desired trajectory. The leader is unaware of the followers, and focuses on reaching the desired trajectory. Noise or unmodeled dynamics may cause the leader to never reach this trajectory [7], but at least the synchronization goal $\boldsymbol{q}_f, \dot{\boldsymbol{q}}_f \rightarrow \boldsymbol{q}_l, \dot{\boldsymbol{q}}_l$ is accomplished by the input torques $\boldsymbol{\tau}_f$ of the followers.

2.2.2 Cooperative control

Another synchronization concept is cooperative control, also known as mutual synchronization [7]. A multi-robot system consisting of p fully actuated robot manipulators with n rigid joints is considered.

A common desired trajectory $\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d$ for all the robots is supplied. None of the n robots are assigned as leader, and all of them work together to achieve synchronization and trajectory tracking. The dynamic model of each robot is assumed to be known.

The goal of a cooperative control scheme is to design control laws $\boldsymbol{\tau}_i$ for all the robots in the system, such that the joint positions and velocities $\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i$ $(i = 1, \dots p)$ of the robots are synchronized with both the common desired trajectory $\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d$, and to the joints of the other robots $\boldsymbol{q}_j, \dot{\boldsymbol{q}}_j$ $(j = 1, \dots p, j \neq i)$. The concept is illustrated in Figure 2.2.

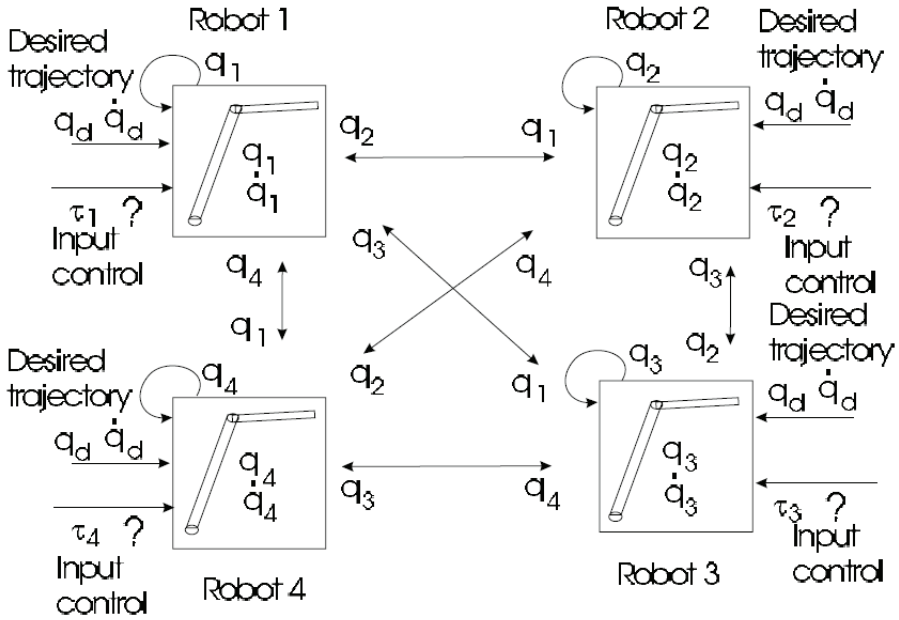

Figure 2.2: Cooperation control. Figure from [7].

Chapter 3

Force Control

Pure position control is sufficient for tasks not involving direct contact between the robot manipulator and the environment. When the end-effector is supposed to polish, grind or apply a force to objects in the workspace in some other fashion, controlling the exerted force is advantageous. In this way the environment, and the end-effector itself, can avoid damage caused by excessive force. Generally there are two different approaches towards force control: *impedance control* and *hybrid control* [14].

3.1 Impedance control

When it is beneficial to model the environment as a deformable or movable object, *impedance control* can be used to control both position of the end-effector and the force it applies to the object. Usually, the environment is modeled as a mass-spring-damper system interacting with the robot manipulator.

The position of the end-effector and the applied force are directly linked to each other, and the equations describing the dynamics of the system consist of a set of ordinary differential equations: the dynamics of the manipulator charged with a mechanical impedance from the environment [11]. Impedance control is a way of controlling both the position of the manipulator end-effector and the force applied to the environment by adjusting the mechanical impedance of the system. By using joint actuators, the apparent inertia, damping and stiffness can be changed to the desired values.

Consider a simple example where a mass m is placed on a frictionless surface with a spring and a damper connected, and under the influence of an external force F and control input u. The equation of motion is given by

$$m\ddot{x} + d\dot{x} + kx = u + F \tag{3.1}$$

where d is the damping coefficient and k is the spring coefficient. This simple one-degree-of-freedom example is shown in Figure 3.1. By using $u = u(x, \dot{x}, \ddot{x}, F)$ the mechanical impedance of the system can be adjusted to suit the task. For example, by choosing the control input

$$u = F + d\dot{x} - kx$$

and insert into (3.1), the closed-loop system becomes

$$\frac{m}{2}\ddot{x} + kx = F \tag{3.2}$$

We see from (3.2) that the damping of the original system (3.1) is gone, and the mass is apparently halved.

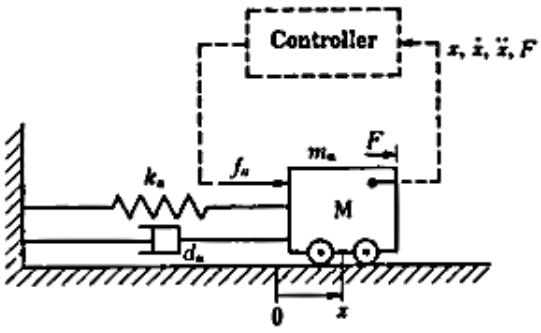

Figure 3.1: Impedance control of 1 DOF system. Figure from [14].

Applications for impedance control includes contact with deformable objects, such as polishing the hood of a car or gripping soft objects, and working on movable objects in the workspace, for example assembling of parts on a rig hanging from the ceiling.

With impedance control one cannot directly control the magnitude of the force exerted on the environment, just the mechanical impedance of the robot-environment system. Therefore, this control scheme will not be used in this thesis. For further information about the impedance control, see [14].

3.2 Hybrid force/motion control

If the environment is modeled as a rigid object, the contact between the end-effector and the surface can be described by algebraic equations. Then the dynamics of the constrained manipulator is given by a differential-algebraic system [11]. The position of the end-effector and the contact force applied to the environment are not directly coupled, and should be tracked independently. This is commonly known as *hybrid force/motion control.*

3.2.1 Natural and artificial constraints

A manipulation task can be decomposed to a set of contact surfaces. Each one of these surfaces is associated with a set of constraints, called *natural constraints*, that follow from the geometry of the task. These constraints can further be divided into position constraints and force constraints. A manipulator not being able to move through a rigid surface is a natural position constraint. Since it cannot apply arbitrary forces tangent to a frictionless surface, this counts as a natural force constraint. The natural position constraints go along the normals to the surface, while the natural force constraints go along the tangents.

The user specifies *artificial constraints* in addition to the natural ones to make the manipulator follow the desired trajectories in position or force. As opposed to the natural constraints, the artificial position constraints act along surface tangents, and artificial force constraints along the surface normals. A comprehensive study of these constraints can be found in Mason [6].

A simple example of a force control task is the insertion of a peg into a hole with constant speed v_d, as illustrated by Figure 3.2. The partially inserted peg cannot move in the x and y direction, and cannot rotate around these axes either. If frictionless surfaces are assumed, arbitrary force along and torque around the z axis cannot be applied. When these natural constraint are found, the artificial constraints are defined to accomplish the goal of inserting the peg: zero force along and torque around the x and y axes, zero rotation around the z axis and a vertical speed of v_d.

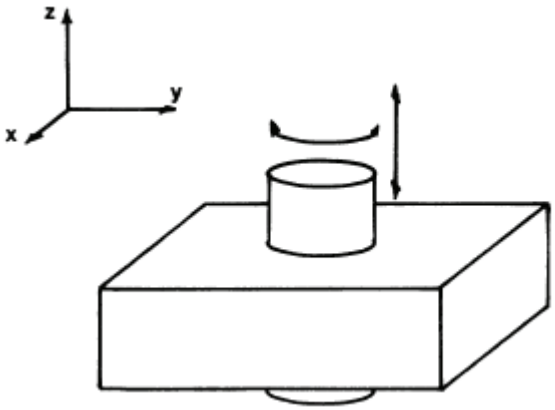

Figure 3.2: Insertion of peg in hole. Figure from [6].

By defining the velocity vector

$$\boldsymbol{v} = \begin{bmatrix} v_x, & v_y, & v_z, & \omega_x, & \omega_y, & \omega_z \end{bmatrix}^{\mathrm{T}} \tag{3.3}$$

and the force vector

$$\boldsymbol{F} = \begin{bmatrix} f_x, & f_y, & f_z, & \tau_x, & \tau_y, & \tau_z \end{bmatrix}^{\mathrm{T}} \tag{3.4}$$

the natural and artificial constraints of the peg-in-hole task can be stated in the constraint frame as shown in Table 3.1.

Table 3.1: Constraints for the peg-in-hole task

Natural	Artificial
$v_x = 0$	$f_x = 0$
$v_y = 0$	$f_y = 0$
$f_z = 0$	$v_z = v_d$
$\omega_x = 0$	$\tau_x = 0$
$\omega_y = 0$	$\tau_y = 0$
$\tau_z = 0$	$\omega_z = 0$

3.2.2 Raibert and Craig: The hybrid controller

To be able to control the position of the manipulator end-effector in some directions, and the force applied in other directions, the hybrid force/motion control concept was proposed by Raibert and Craig [9].

The goal of Raibert and Craig was to present a conceptually simple scheme to control both the position of the end-effector and the force exerted on the environment at the same time by combining force measurements from wrist-mounted force sensors and position data from the joints.

The description of the end-effector constraints is based on the Mason's work on natural and artificial constraints [6], as mentioned in the previous section. A Cartesian coordinate frame, the constraint frame $\{C\}$, is defined, and is conveniently placed according to the task geometry. In this way, the position and force constraints will align with the axes of $\{C\}$, and the constraints and trajectories are easily specified.

Figure 3.3 shows an example where a crank is turned by a robot manipulator. The constraint frame is attached to the crank, thus the force constraint expressed in $\{C\}$ does not change as the handle turns.

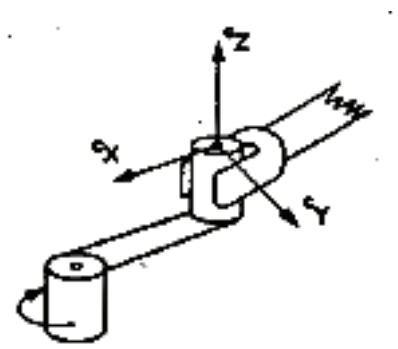

Figure 3.3: A manipulator turning a crank. Figure from [9].

Before the separation of position and force controlled degrees of freedom can commence, position and force measurements must be transformed to the constraint frame. The position of the end-effector expressed in the constraint frame, $\boldsymbol{X} \in \mathbb{R}^6$, is related to the joint coordinates $\boldsymbol{q} \in \mathbb{R}^6$ by the forward kinematic transform $\boldsymbol{\Lambda}$:

$$\boldsymbol{X}^C = \boldsymbol{\Lambda}(\boldsymbol{q})$$

The relation between the time derivative of Cartesian coordinates of the end-effector, $\dot{\boldsymbol{X}}^C$ and the joint velocities $\dot{\boldsymbol{q}}$ is expressed using the manipulator Jacobian $\mathbf{J}(\boldsymbol{q})$ [8]:

$$\dot{\boldsymbol{X}}^C = \mathbf{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

The forces measured by the end-effector sensor are transformed into the constraint frame by the transformation matrix $\mathbf{T}_H^C$

$$\boldsymbol{F}^C = (\mathbf{T}_H^C)\boldsymbol{F}^H$$

Then the errors in position, velocity and force in the constraint frame are defined as

$$\begin{aligned}
\Delta\boldsymbol{X} &\triangleq \boldsymbol{X}_d^C - \boldsymbol{X}^C = \boldsymbol{X}_d^C - \boldsymbol{\Lambda}(\boldsymbol{q}) \\
\Delta\dot{\boldsymbol{X}} &\triangleq \dot{\boldsymbol{X}}_d^C - \dot{\boldsymbol{X}}^C = \dot{\boldsymbol{X}}_d^C - \mathbf{J}\dot{\boldsymbol{q}} \\
\Delta\boldsymbol{F} &\triangleq \boldsymbol{F}_d^C - \boldsymbol{F}^C
\end{aligned}$$

where $\boldsymbol{X}_d^C$ is the position reference for the end-effector. Now that the errors are expressed in the constraint frame, they can be separated into position and force controlled degrees of freedom by using the compliance selection matrix $\mathbf{S}$ defined in [9] as

$$\mathbf{S} \triangleq \operatorname{diag}(\begin{bmatrix} s_1, & s_2, & s_3, & s_4, & s_5, & s_6 \end{bmatrix}^{\mathrm{T}}) \tag{3.5}$$

where $s_i = 0$ indicates position control, and $s_i = 1$ indicates force control. As the task geometry and natural constraints change, so will the selection matrix.

For example, for the peg-in-hole example in Figure 3.2, the selection matrix will become

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Using the selection matrix (3.5), the separated errors are defined as

$$\begin{aligned} \boldsymbol{X}_e^C &\triangleq (\mathbf{I} - \mathbf{S})\Delta\boldsymbol{X} = (\mathbf{I} - \mathbf{S})(\boldsymbol{X}_d^C - \boldsymbol{\Lambda}(\boldsymbol{q})) \\ \dot{\boldsymbol{X}}_e^C &\triangleq (\mathbf{I} - \mathbf{S})\Delta\dot{\boldsymbol{X}} = (\mathbf{I} - \mathbf{S})(\dot{\boldsymbol{X}}_d^C - \mathbf{J}\dot{\boldsymbol{q}}) \\ \boldsymbol{F}_e^C &\triangleq \mathbf{S}\Delta\boldsymbol{F} = \mathbf{S}(\boldsymbol{F}_d^C - \boldsymbol{F}^C) \end{aligned}$$

Once separated, the position and force errors are transformed back to the joint space and used as inputs for the hybrid position/force controller. The transformations are given by

$$\boldsymbol{q}_e = \mathbf{J}^{-1}(\boldsymbol{q})\boldsymbol{X}_e^C \tag{3.6}$$

$$\dot{\boldsymbol{q}}_e = \mathbf{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{X}}_e^C \tag{3.7}$$

$$\boldsymbol{\tau}_e = \mathbf{J}^{\mathrm{T}}(\boldsymbol{q})\boldsymbol{F}_e^C \tag{3.8}$$

where the inverse Jacobian is used to transform position and velocity, and the Jacobian transpose transforms the force. Equation (3.6) is an approximation valid for small $\boldsymbol{X}_e^C$ [9]. The inverse of the Jacobian is one of the more computationally demanding steps of this algorithm. It is assumed that the square 6×6 Jacobian matrix is nonsingular; thus the inverse exists.

The complete controller is given by

$$\boldsymbol{\tau} = \boldsymbol{\tau}_p + \boldsymbol{\tau}_f \tag{3.9}$$

$$\boldsymbol{\tau}_p = \boldsymbol{\tau}_p(\boldsymbol{q}_e, \dot{\boldsymbol{q}}_e) \tag{3.10}$$

$$\boldsymbol{\tau}_f = \boldsymbol{\tau}_f(\boldsymbol{\tau}_e) \tag{3.11}$$

Raibert and Craig suggest the addition of feedforward compensation terms for the nonlinear dynamics of robot manipulators, but they do not pursue the subject further.

Figure 3.4 illustrates the hybrid position/force controller. The controller consists of the sum of two terms: the position feedback loop $\boldsymbol{\tau}_p$ (upper half of the illustration) and the force feedback loop $\boldsymbol{\tau}_f$ (lower half).

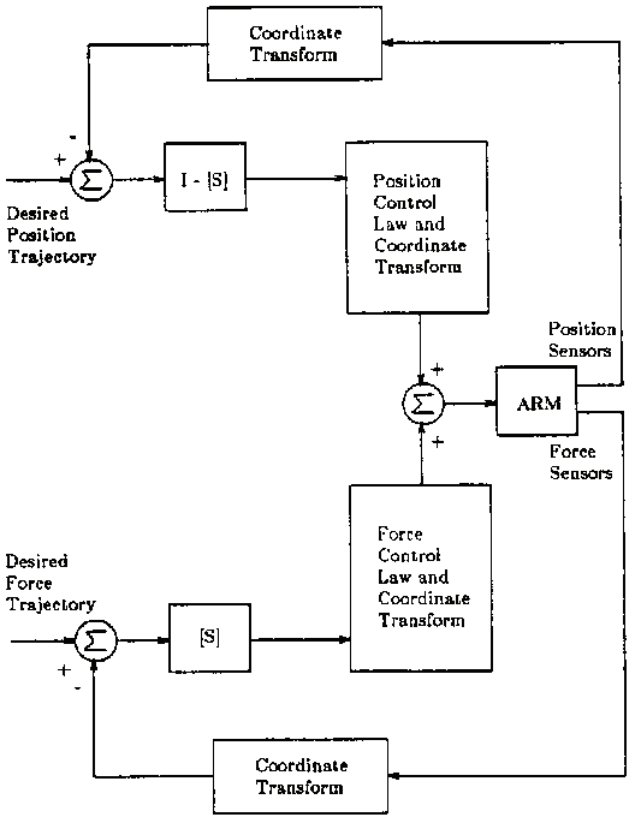

Figure 3.4: Hybrid controller based on approach in [9]. Figure from [12].

The two separate control laws will both affect each joints of the robot manipulator; each joint contributes to the arm motion and the forces applied simultaneously. This is in contrast with some earlier control algorithms where a joint is either position or force controlled [12]. The desired position in the direction normal to the surface will not act as a disturbance in the case of an inaccurate surface, because the multiplication of the position selection matrix $(\mathbf{I} - \mathbf{S})$ will set this error to zero. This hybrid approach does not specify the design of the position and force control laws; it is only supplying the framework.

The use of orthogonal artificial and natural constraints is necessary to make the velocity constraints produce the desired trajectory. In addition, an error in the planned natural constraints combined with non-orthogonal artificial force constraints may cause large actuator forces and/or loss of surface contact [6].

As shown in Figure 3.4 and the text above, the hybrid force/motion concept Raibert and Craig introduced in [9] is intuitive, and the separation of the degrees of freedom is easily done once the constraints and the measurements are expressed in the constraint frame. However, the coordinate transformations involved can be quite comprehensive [12].

Another issue is that the manipulator dynamics needs to be taken into account rigorously, and because of this it is problematic to show theoretically that the desired position and force trajectories can be realized simultaneously [13]. Therefore, the articles involving Raibert and Craig's hybrid approach tend to not contain mathematical proofs of stability; they verify stability by simulation and experiments [9] [12].

3.2.3 Dynamic hybrid control

Yoshikawa proposed the dynamic hybrid control method in [13]. Yoshikawa acknowledge that the method of Raibert and Craig [9] described in the previous section is "conceptually much clearer than previous ones and seems to have good possibility of practical use". On the other hand, Raibert and Craig is criticized for not taking the manipulator dynamics into consideration, and therefore it has not been shown theoretically that it is possible to track both position and force trajectories at the same time. Yoshikawa also dismisses the importance of Mason's [6] work on task constraints related to the dynamics of constrained motion of robot manipulators.

The new hybrid control method is based on the dynamic equations of robot manipulators and constraints on the end-effector to ensure contact with the environment. The constraints are given as a set of hypersurfaces expressed in the end-effector coordinates. Together these equations give the joint driving torque needed for realizing both the position and force trajectory. Different tasks, such as part assembly or surface tracing, will constrain various degrees of freedom. Mason [6] did also discuss the use of hypersurfaces to describe end-effector constraints, but he did not consider the acceleration of the manipulator; only the kinematics.

In this section I will use the notation from Yoshikawa, and will go through the most important steps. For more details, see [13].

Let $\boldsymbol{r} \in \mathbb{R}^6$ be the end-effector position and orientation vector relative to a fixed frame. Then the constraints on the manipulator can be expressed by the set of m hypersurfaces

$$p_i(\boldsymbol{r}) = 0, \qquad i = 1, 2, \cdots, m \tag{3.12}$$

where $m \leq 6$ and p_i is assumed to be mutually independent and twice differentiable. The time derivatives of (3.12) are

$$\mathbf{E}_F \dot{\boldsymbol{r}} = \mathbf{0} \tag{3.13}$$

$$\mathbf{E}_F \ddot{\boldsymbol{r}} + \dot{\mathbf{E}}_F \dot{\boldsymbol{r}} = \mathbf{0} \tag{3.14}$$

where the rows of $\mathbf{E}_F$ are of unit length.

Then, a set of $6-m$ unit-length vectors are chosen such that they are differentiable functions of r, mutually independent and independent of the rows of $\mathbf{E}_F$. Let these vectors be the rows of the matrix $\mathbf{E}_P$ and

$$\mathbf{E} = \begin{bmatrix} \mathbf{E}_P \\ \mathbf{E}_F \end{bmatrix} \tag{3.15}$$

The coordinate system with its origin at the present end-effector position $\boldsymbol{r}$ and with the basis given by the row vectors of $\mathbf{E}$ can be named the constraint frame. Observe that this frame is not necessarily orthogonal, as it is in the case of Raibert and Craig [9]. This is dependent on the constraints given by (3.12) and the vectors chosen in $\mathbf{E}_P$. Usually it is desirable to make the basis orthogonal, but complex constraints may make this hard to obtain.

The rows of $\mathbf{E}_F$ represents the coordinate axes normal to the constraint surface, along which artificial force constraints [6] should be specified. $\mathbf{E}_P$ expresses the axes which complement $\mathbf{E}_F$, and should be used to specify artificial position constraints. $\mathbf{E}$ is the rotation matrix from the inertial frame to the constraint frame.

Figure 3.5 shows a simple planar two-degree-of-freedom manipulator together with a constraint surface. The normal vector of the surface gives $\mathbf{E}_F^{\mathrm{T}}$, and the tangent vector gives $\mathbf{E}_P^{\mathrm{T}}$.

Figure 3.5: Manipulator with constraint surface. Figure from [13].

The velocity and acceleration of the end-effector expressed in the constraint frame are given by

$$\mathbf{E}\dot{r} = \begin{bmatrix} \mathbf{E}_P\dot{\boldsymbol{r}} \\ \mathbf{0} \end{bmatrix} \tag{3.16}$$

$$\mathbf{E}\ddot{r} = \begin{bmatrix} \mathbf{E}_P\ddot{\boldsymbol{r}} \\ -\dot{\mathbf{E}}_F\dot{\boldsymbol{r}} \end{bmatrix} \tag{3.17}$$

which implies that the normal component of the end-effector velocity relative to the surface is zero.

Consider a robot manipulator with $n \geq 6$ degrees of freedom with the dynamics given by

$$\mathbf{M}\ddot{\boldsymbol{q}} + \mathbf{C}\dot{\boldsymbol{q}} + \mathbf{D}\dot{\boldsymbol{q}} + \boldsymbol{g} = \boldsymbol{\tau}_c - \mathbf{J}^{\mathrm{T}}\mathbf{E}_F^{\mathrm{T}}\boldsymbol{f}_F$$

where $\mathbf{J}^{\mathrm{T}}\mathbf{E}_F^{\mathrm{T}}\boldsymbol{f}_F$ is the torque caused by the constraining force. Letting $\ddot{\boldsymbol{r}}_{EPd} \in \mathbb{R}^{6-m}$ be the desired acceleration and $\boldsymbol{f}_{Fd} \in \mathbb{R}^m$ be the desired force, Yoshikawa found the following equations for the joint driving force command $\boldsymbol{\tau}_c$, :

$$\boldsymbol{\tau}_c = \boldsymbol{\tau}_P + \boldsymbol{\tau}_F \tag{3.18}$$

$$\boldsymbol{\tau}_P = \mathbf{M}\ddot{\boldsymbol{q}}_d + \mathbf{C}\dot{\boldsymbol{q}} + \mathbf{D}\dot{\boldsymbol{q}} + \boldsymbol{g} \tag{3.19}$$

$$\boldsymbol{\tau}_F = \mathbf{J}^{\mathrm{T}}\mathbf{E}_F^{\mathrm{T}}\boldsymbol{f}_{Fd} \tag{3.20}$$

$$\ddot{\boldsymbol{q}}_d = \mathbf{J}^{+}\left\{\mathbf{E}^{-1}\begin{bmatrix} \ddot{\boldsymbol{r}}_{EPd} \\ -\dot{\mathbf{E}}_F\dot{\boldsymbol{r}} \end{bmatrix} - \dot{\mathbf{J}}\dot{\boldsymbol{q}}\right\} + (\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\boldsymbol{k} \tag{3.21}$$

where $\mathbf{J}^{+}$ is the pseudoinverse of the Jacobian $\mathbf{J}$, and the vector $\boldsymbol{k}$ is an arbitrary time function when considering redundant robot manipulators, $n > 6$. As in the hybrid control scheme proposed by Raibert and Craig [9], this controller is composed of the sum of two forces; one for controlling the position of the end-effector, $\boldsymbol{\tau}_P$, and one for controlling the force, $\boldsymbol{\tau}_F$. By assuming that rank $\mathbf{J} = 6$, i.e. the manipulator is not in a singular position, it can be shown that $\ddot{\boldsymbol{r}}_{EP} = \ddot{\boldsymbol{r}}_{EPd}$ and $\boldsymbol{f}_F = \boldsymbol{f}_{Fd}$, and both position and force tracking is achieved.

If the constraint hypersurfaces given by (3.12) or the manipulator equations are not exactly correct, the open loop controller given by (3.18) - (3.21) will not suffice, and the system will deviate from the desired trajectories. The solution is to add feedback from the measured position and force as shown in Figure 3.6.

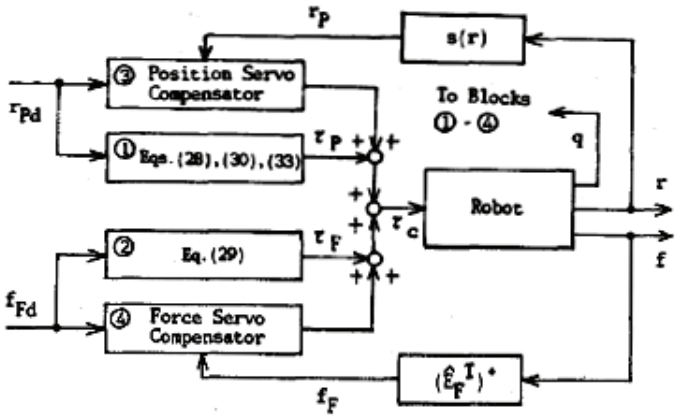

Figure 3.6: Dynamic hybrid controller based on approach in [13].

The setup in Figure 3.6 differs from Raibert and Craig [9] with having feedforward terms compensating for the dynamics of the manipulator, given by (3.19) and (3.21). Yoshikawa [13] shows the importance of the feedforward term $\boldsymbol{f}_{Fd}$ when dynamic control of position and force is needed. On the other hand, Raibert and Craig [9] included this term on the assumption of quasi-static operation.

Chapter 4

The Proposed Control Concept

4.1 Joint space or task space control

The robot manipulator's reference is a trajectory specifying the end-effector position, velocity and acceleration. The motion control of the robot can be performed in *joint space*, given by the joint angles $\boldsymbol{q}$, or in *task space*, represented by the vector $\boldsymbol{X}$ containing the end-effector position relative to an inertial frame.

Control in joint space requires the use of inverse kinematics of the robot manipulator to transform the reference trajectory from task space to joint space. The exception is when the reference already is given in joint space, such as in the case when the states of leader manipulator is used as a reference. When the coordinates are transformed, a control algorithm is applied to the joint errors $\boldsymbol{q}_e$. Closed-form inverse kinematics equations are favorable, because they need to be solved each time step, and an iterative algorithm is more time consuming. These equations are generally nonlinear functions of the joint variables $\boldsymbol{q}$, and the derivation of these closed-form equations may be difficult depending on the kinematic structure of the manipulator and the number of degrees of freedom [10].

The inverse kinematics problem may have no solution, for example when the desired point is out of reach of the robot arm. When a solution exists, it may not be unique. The robot can reach a point in different ways, for example with the elbow of the manipulator in the up- or down-position. Having closed-form solutions will also make room for a way to choose a solution when more than one is available. Hence, one advantage of joint space control

is the ability to choose the direction of the links of the robot, and can be of help when trying to avoid collision with the environment.

If one chooses to control the motion of the robot manipulator in task space, the need for inverse kinematics is not present [1]. The controller closes the feedback loop directly on the end-effector position and velocity given in task space. According to Caccavale et al. [1] is joint space suited for simple robot tasks, but complex tasks involving interaction with the environment requires task space control.

Instead of transforming the desired trajectory from task space to joint space by the means of inverse kinematics, we now transform the measurements from joint space to task space by the forward kinematics. The forward kinematics is generally less complex than the inverse problem, and can be found using the Denavit-Hartenberg convention [10].

I chose to control the position and velocity of the manipulator end-effector in task space. It is easier to transform the coordinates of the end-effector from joint space to task space by the forward kinematics than the other way around, and the decoupling of the force controlled and position controlled degrees of freedom will happen in task space.

The forward kinematics is given by the homogeneous transformation

$$\mathbf{T}(\boldsymbol{q}) = \begin{bmatrix} \mathbf{R}(\boldsymbol{q}) & \boldsymbol{p}(\boldsymbol{q}) \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3) \tag{4.1}$$

where $\mathbf{R} \in SO(3)$ is the rotation matrix and $\boldsymbol{p} \in \mathbb{R}^3$ is the Cartesian position of the end-effector, both functions of the joint angles $\boldsymbol{q}$ and expressed relative to an inertial frame.

The differential kinematics relates the joint velocities $\dot{\boldsymbol{q}}$ to the end-effector linear and angular velocities $\boldsymbol{v}$:

$$\boldsymbol{v} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4.2}$$

where $\mathbf{J}$ is the end-effector geometric Jacobian Matrix[1].

[1] See Spong et al. [10] for the calculation of the Jacobian matrix.

By taking the time derivative of (4.2) we obtain the relationship between the acceleration of the end-effector $\dot{\boldsymbol{v}}$ and the joint accelerations $\ddot{\boldsymbol{q}}$:

$$\dot{\boldsymbol{v}} = \begin{bmatrix} \ddot{\boldsymbol{p}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \mathbf{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\mathbf{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4.3}$$

Joint configurations causing the geometric Jacobian to lose full rank are called kinematic singularities. This can happen on the boundary of the manipulator workspace restricting motion in certain directions, or when the joints are lined up such that a bounded end-effector velocity may be caused by unbounded joint velocities. For example, when two rotating axes are aligned, they can rotate in opposite direction with arbitrary velocity, but the end-effector remains at rest.

In this thesis it is assumed that the robot manipulator avoids kinematic singularities, thus the geometric Jacobian $\mathbf{J}$ has always full rank.

4.2 Choice of orientation representation

After the choice of task space control, the issue of how to represent the end-effector orientation must be addressed. Three common representations[2] are

1. Rotation matrix: $\mathbf{R} \in \mathbb{R}^{3\times3}$
2. Euler angles : $\mathbf{\Theta} = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3$
3. Unit quaternions: $\boldsymbol{q} = \begin{bmatrix}\eta & \epsilon_1 & \epsilon_2 & \epsilon_3\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^4$

One advantage of using a rotation matrix to represent the orientation of the manipulator end-effector is that the forward kinematics $\mathbf{T}$, given by (4.1), already contains the orientation in the form of a rotation matrix $\mathbf{R}$, requiring no further transformation. In addition, this representation is free of singularities. $\mathbf{R}$ contains 9 elements, far more than a minimal representation, with 6 orthonormality constraints, making it a redundant representation. In calculations, because of numerical errors, an algorithm to ensure the preservation of the orthonormality constraints is needed.

Euler angles are a minimal representation using only 3 parameters to represent a rotation. They are quite intuitive, consisting of three rotations around the principal axes. Of the 12 possible Euler angle definitions, Roll-Pitch-Yaw (ZYX) seems to be the most appropriate one. The drawback of Euler angles is the representation singularities; for Roll-Pitch-Yaw this occurs at $\theta = \pm\frac{\pi}{2}$. No post-processing is needed to make sure they stay Euler angles.

The great advantage of unit quaternions is that they are a singularity-free orientation representation using only 4 parameters. Thus, this is no no minimal representation, and a length constraint is imposed on them. Quaternions are not as intuitive as Euler angles, and every rotation has two quaternion representations with opposite signs.

Based on the above, I chose to use Euler angles (Roll-Pitch-Yaw) to represent the orientation of the end-effector of the robot manipulator in this thesis. They are easy to interpret, and they make it possible to separate position and force controlled degrees of freedom of the orientation. In addition, because this is a minimal representation, the analytical Jacobian becomes square when considering robot manipulators with $n = 6$ joints.

[2] For more on different orientation representation, see [3].

The Euler angles $\boldsymbol{\Theta} = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3$ represent a rotation consisting of three successive rotations around the principal axes. By first rotating an angle ψ around the inertial x-axis, then an angle θ around the inertial y-axis, and finally an angle ϕ around the inertial z-axis, the total rotation matrix is given by

$$\begin{aligned}\mathbf{R}(\boldsymbol{\Theta}) &= \mathbf{R}_z(\phi)\mathbf{R}_y(\theta)\mathbf{R}_x(\psi) \\ &= \begin{bmatrix}\mathrm{c}\phi & -\mathrm{s}\phi & 0\\ \mathrm{s}\phi & \mathrm{c}\phi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\mathrm{c}\theta & 0 & \mathrm{s}\theta\\ 0 & 1 & 0\\ -\mathrm{s}\theta & 0 & \mathrm{c}\theta\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & \mathrm{c}\psi & -\mathrm{s}\psi\\ 0 & \mathrm{s}\psi & \mathrm{c}\psi\end{bmatrix} \\ &= \begin{bmatrix}\mathrm{c}\phi\,\mathrm{c}\theta & -\mathrm{s}\phi\,\mathrm{c}\psi + \mathrm{c}\phi\,\mathrm{s}\theta\,\mathrm{s}\psi & \mathrm{s}\phi\,\mathrm{s}\psi + \mathrm{c}\phi\,\mathrm{s}\theta\,\mathrm{c}\psi\\ \mathrm{s}\phi\,\mathrm{c}\theta & \mathrm{c}\phi\,\mathrm{c}\psi + \mathrm{s}\phi\,\mathrm{s}\theta\,\mathrm{s}\psi & -\mathrm{c}\phi\,\mathrm{s}\psi + \mathrm{s}\phi\,\mathrm{s}\theta\,\mathrm{c}\psi\\ -\mathrm{s}\theta & \mathrm{c}\theta\,\mathrm{s}\psi & \mathrm{c}\theta\,\mathrm{c}\psi\end{bmatrix}\end{aligned} \tag{4.4}$$

where $\mathrm{c}i, \mathrm{s}j$ is short for $\cos i, \sin j$ respectively. By comparing the elements of (4.4) with the elements of the rotation matrix, $r(i,j)$, the Euler angles can be extracted using trigonometric identities.

The relationship between the time derivative of the Euler angles, $\dot{\boldsymbol{\Theta}}$, and the angular velocity $\boldsymbol{\omega}$ is derived using the the time derivative of (4.4). By comparing this result with the well-known equation

$$\dot{\mathbf{R}} = \mathbf{S}(\boldsymbol{\omega})\mathbf{R}$$

where $\mathbf{S}$ is the skew symmetric matrix defined by the angular velocity $\boldsymbol{\omega}$, the relation becomes

$$\boldsymbol{\omega} = \mathbf{B}(\boldsymbol{\Theta})\dot{\boldsymbol{\Theta}} \tag{4.5}$$

where

$$\mathbf{B}(\boldsymbol{\Theta}) = \begin{bmatrix}0 & -\mathrm{s}\phi & \mathrm{c}\phi\,\mathrm{c}\theta\\ 0 & \mathrm{c}\phi & \mathrm{s}\phi\,\mathrm{c}\theta\\ 1 & 0 & -\mathrm{s}\theta\end{bmatrix} \tag{4.6}$$

4.2.1 Singularities

The matrix $\mathbf{B}$ giving the transformation between the time derivative of Euler angles $\dot{\boldsymbol{\Theta}}$ and the angular velocity $\boldsymbol{\omega}$, is the cause of the representational singularities introduced by the Euler angles. $\mathbf{B}$, given by (4.6), becomes singular when $\theta = \pm\frac{\pi}{2}$.

When the end-effector orientation becomes close to a singularity, the extraction of the Euler angles $\boldsymbol{\Theta}$ from the rotation matrix $\mathbf{R}$ becomes ill-conditioned. One way to avoid the singularities caused by the chosen Euler angle definition is to change to another Euler angle definition when close to a singularity of the first one. Then by switching between these two representations, the singularities can be avoided.

An alternative definition of the orientation error based on Euler angles is proposed by Caccavale et al. [1]. Instead of using the actual rotation matrix for basis for the Euler angles, one can use the matrix describing the rotation between the desired and the actual end-effector orientation:

$$\mathbf{R}_e = \mathbf{R}^{\mathrm{T}}\mathbf{R}_d \tag{4.7}$$

where $\mathbf{R}_d$ is the desired orientation, $\mathbf{R}$ is the current end-effector orientation and $\mathbf{R}_e$ is the rotation between them. By extracting the Euler angles from (4.7) instead of directly from $\mathbf{R}$ and $\mathbf{R}_d$ one avoids the singularities as long the angle error in θ is less than $\pm\frac{\pi}{2}$. The matrix $\mathbf{B}$ will not be ill-conditioned because the of the actual or desired end-effector orientation being close to a singularity, but only if the orientation error in θ is large. If the initial orientation is not far off from the desired orientation, and a convergent tracking algorithm is applied, this scheme will perform better [1].

4.2.2 Analytical Jacobian

The Jacobian matrix $\mathbf{J}(\boldsymbol{q})$ used in (4.2) is called the *geometric Jacobian*, and transforms joint velocities $\dot{\boldsymbol{q}}$ into linear and angular velocities $\dot{\boldsymbol{p}}, \boldsymbol{\omega}$. When using a minimal representation of the orientation of the manipulator end-effector, it is desirable to transform the joint velocities to the time derivative of position and Euler angles.

Let the vector

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{\Theta} \end{bmatrix} \tag{4.8}$$

denote the position $\boldsymbol{p}$ and Euler angles $\boldsymbol{\Theta}$ of the end-effector. We want to find the relation

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{\Theta}} \end{bmatrix} = \mathbf{J}_a(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4.9}$$

where $\mathbf{J}_a$ is called the *analytical Jacobian* [10]. The time derivative of (4.9) gives the transformation from joint accelerations $\ddot{\boldsymbol{q}}$ to linear and Euler angle accelerations $\ddot{\boldsymbol{p}}, \ddot{\boldsymbol{\Theta}}$:

$$\ddot{\boldsymbol{X}} = \begin{bmatrix} \ddot{\boldsymbol{p}} \\ \ddot{\boldsymbol{\Theta}} \end{bmatrix} = \mathbf{J}_a(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\mathbf{J}}_a(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4.10}$$

By using (4.2) and (4.5), the relation between the geometric and analytical Jacobian is found to be

$$\begin{aligned} \mathbf{J}(\boldsymbol{q})\dot{\boldsymbol{q}} &= \begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \mathbf{B}(\boldsymbol{\Theta})\dot{\boldsymbol{\Theta}} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}(\boldsymbol{\Theta}) \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{\Theta}} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}(\boldsymbol{\Theta}) \end{bmatrix} \mathbf{J}_a(\boldsymbol{q})\dot{\boldsymbol{q}} \end{aligned}$$

Hence, the analytical Jacobian is expressed by

$$\mathbf{J}_a(\boldsymbol{q}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^{-1}(\boldsymbol{\Theta}) \end{bmatrix} \mathbf{J}(\boldsymbol{q}) \tag{4.11}$$

The singularities of the analytical Jacobian $\mathbf{J}_a(\boldsymbol{q})$ consists of the kinematic singularities included in the geometric Jacobian $\mathbf{J}(\boldsymbol{q})$ and the representational singularities of the transformation matrix $B(\boldsymbol{\Theta})$.

4.3 Design decisions and assumptions

Before the derivation of the hybrid force/motion controller can be presented, the design decisions and assumptions made will be stated.

4.3.1 Design decisions

The hybrid force/motion control concept proposed by Raibert & Craig [9] is intuitive and easy to comprehend. But they did not account for the dynamics of the robot manipulator, and did not prove stability.

On the other hand, the dynamic hybrid control scheme that was proposed by Yoshikawa [13] is shown to be stable in both position and force. The drawback is a more complex controller.

I chose to include the simplicity of the Raibert and Craig separation technique, in combination with feedforward compensation for the nonlinear manipulator dynamics. For simplicity, it is assumed that the constraint frame is constant and aligned with the inertial frame spanning the manipulator task space.

A coordinated control scheme, where one leader guides the other followers, was chosen for the synchronization of the robot manipulators. Then the synchronization task is separated into two subproblems: the leader tracks the desired trajectory with its own tracking controller, and the coordination is accomplished by the followers trying to converge to the leader's trajectory [5].

Coordinated control requires less data transfer between the robots because only the leader's trajectory is given to all the followers. In a cooperative control scheme, none of the robots is assigned as leader and all of them transmit their trajectories to all the other robots [7]. Thus, the coordination controller will depend only on the states of the leader and *one* of the followers, while the cooperative controller must take into account the states of *all* the robots of the multi-robot system.

For simplicity, I chose to use the basic Euler Angle approach, in stead of the more robust scheme proposed in [1]. Throughout this thesis it is assumed that the analytical Jacobian $\mathbf{J}_a$ has full rank. This is equivalent with no kinematic or representational singularities.

4.3.2 Assumptions

It is assumed that both the leader and the followers are fully actuated robots with $n = 6$ rigid joints each. For simplicity, the multi-robot system is considered to be consisting of one leader manipulator and one follower. This is without loss of generality because the followers are synchronized individually in a coordinated control scheme.

The dynamics of the robots is represented by the Euler-Lagrange equations of motion for robot manipulators given by (2.3). The model parameters are assumed to be exactly known, and position, velocity and acceleration of both the leader and follower are measured with no noise present. There is no limit opposed on the input torque, and a force sensor placed in the robot manipulator end-effector measures the applied force $\boldsymbol{F}$.

The leader robot is already driven by an input torque $\boldsymbol{\tau}_l$, that ideally will ensure convergence of the leader's joint positions and velocities $\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l$ to the desired trajectory $\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d$. Thus, the remaining task is to create a controller $\boldsymbol{\tau}$ to ensure that the follower converges to the leader's trajectory.

To increase readability in the following derivations, the "(t)" is removed from the time dependent variables, and the configuration dependent matrices and vectors will be written without the "$(\boldsymbol{q})$" and "$(\dot{\boldsymbol{q}})$".

4.4 The proposed hybrid controller

This section will present the derivation of the proposed hybrid force/motion controller.

When the manipulator end-effector is in contact with the environment, the equations of motion in joint space must include the torque caused by the end-effector force $\boldsymbol{F}$ [10]. The dynamics is then given by

$$\mathbf{M}\ddot{\boldsymbol{q}} + \mathbf{C}\dot{\boldsymbol{q}} + \boldsymbol{g} + \mathbf{J}^{\mathrm{T}}\boldsymbol{F} = \boldsymbol{\tau} \tag{4.12}$$

where $\mathbf{J}$ is the geometric Jacobian mapping the forces acting at the end-effector to torques at the manipulator joints. Then we use the control input $\boldsymbol{\tau}$ to create a modified inverse dynamics[3] control law to compensate for the nonlinear dynamics of the manipulator:

$$\boldsymbol{\tau} = \mathbf{C}\dot{\boldsymbol{q}} + \boldsymbol{g} + \mathbf{M}\mathbf{J}_a^{-1}(\boldsymbol{u} - \dot{\mathbf{J}}_a\dot{\boldsymbol{q}}) + \mathbf{J}^{\mathrm{T}}\boldsymbol{u}_f \tag{4.13}$$

where $\mathbf{J}_a$ is the analytical Jacobian given by (4.11) and $\boldsymbol{u}, \boldsymbol{u}_f \in \mathbb{R}^6$ are new input variables yet to be chosen. By inserting the controller given by (4.13) into the robot equation (4.12) the system becomes

$$\begin{aligned} \mathbf{M}\ddot{\boldsymbol{q}} + \mathbf{C}\dot{\boldsymbol{q}} + \boldsymbol{g} + \mathbf{J}^{\mathrm{T}}\boldsymbol{F} &= \mathbf{C}\dot{\boldsymbol{q}} + \boldsymbol{g} + \mathbf{M}\mathbf{J}_a^{-1}(\boldsymbol{u} - \dot{\mathbf{J}}_a\dot{\boldsymbol{q}}) + \mathbf{J}^{\mathrm{T}}\boldsymbol{u}_f \\ \mathbf{M}\ddot{\boldsymbol{q}} &= \mathbf{M}\mathbf{J}_a^{-1}(\boldsymbol{u} - \dot{\mathbf{J}}_a\dot{\boldsymbol{q}}) + \mathbf{J}^{\mathrm{T}}(\boldsymbol{u}_f - \boldsymbol{F}) \end{aligned} \tag{4.14}$$

Ultimately, we want to express the manipulator dynamics in task space, $\ddot{\boldsymbol{X}}$. Then the position controlled and force controlled degrees of freedom can easily be separated. Further manipulation of (4.14) gives

$$\begin{aligned} \ddot{\boldsymbol{q}} &= \mathbf{J}_a^{-1}(\boldsymbol{u} - \dot{\mathbf{J}}_a\dot{\boldsymbol{q}}) + \mathbf{M}^{-1}\mathbf{J}^{\mathrm{T}}(\boldsymbol{u}_f - \boldsymbol{F}) \\ \mathbf{J}_a\ddot{\boldsymbol{q}} &= \boldsymbol{u} - \dot{\mathbf{J}}_a\dot{\boldsymbol{q}} + \mathbf{J}_a\mathbf{M}^{-1}\mathbf{J}^{\mathrm{T}}(\boldsymbol{u}_f - \boldsymbol{F}) \\ \mathbf{J}_a\ddot{\boldsymbol{q}} + \dot{\mathbf{J}}_a\dot{\boldsymbol{q}} &= \boldsymbol{u} + \mathbf{J}_a\mathbf{M}^{-1}\mathbf{J}^{\mathrm{T}}(\boldsymbol{u}_f - \boldsymbol{F}) \end{aligned} \tag{4.15}$$

[3] The method of inverse dynamics is a special case of feedback linearization [10].

Now, by using the relation given by Equation (4.10), (4.15) can be written as

$$\begin{aligned} \ddot{\boldsymbol{X}} &= \boldsymbol{u} + \mathbf{J}_a \mathbf{M}^{-1} \mathbf{J}^{\mathrm{T}} (\boldsymbol{u}_f - \boldsymbol{F}) \\ \ddot{\boldsymbol{X}} &= \boldsymbol{u} + \mathbf{W} (\boldsymbol{u}_f - \boldsymbol{F}) \end{aligned} \tag{4.16}$$

where

$$\mathbf{W} = \mathbf{J}_a \mathbf{M}^{-1} \mathbf{J}^{\mathrm{T}} \tag{4.17}$$

is called the mobility tensor. $\mathbf{W}$ is mapping a force at the end-effector to task space, and has full rank when the analytic Jacobian $\mathbf{J}_a$ has full rank; that is when no kinematic or representational singularities occur.

According to Spong et al. [10], it is often an advantage to assume that $\boldsymbol{u}$ is a pure position and velocity controller and $\boldsymbol{u}_f$ is a pure force controller. In spite of this, they simplify by taking $\boldsymbol{u}_f = \boldsymbol{F}$ to cancel the end-effector force $\boldsymbol{F}$.

The result is the linear and uncoupled double integrator system

$$\ddot{\boldsymbol{X}} = \boldsymbol{u} \tag{4.18}$$

Additional force feedback will be incorporated in $\boldsymbol{u}$, and is possible as long as the the mobility tensor $\mathbf{W}$ has full rank.

Now that the dynamics of the system, given by (4.18), is transformed to the constraint frame, it is time to separate the degrees of freedom into position controlled and force controlled ones. Consider the selection matrices

$$\mathbf{S}_f \triangleq \mathrm{diag}(\begin{bmatrix} s_1, & s_2, & s_3, & s_4, & s_5, & s_6 \end{bmatrix}^{\mathrm{T}}) \tag{4.19}$$

$$\mathbf{S}_p \triangleq \mathbf{I} - \mathbf{S}_f \tag{4.20}$$

where $s_i = 0$ indicates position control, and $s_i = 1$ indicates force control. These matrices are defined in the same way as those in the hybrid concept of Raibert and Craig [9], with a slight notation difference.

Let the error signals be defined as

$$\boldsymbol{X}_e \triangleq \boldsymbol{X}_l - \boldsymbol{X} \tag{4.21}$$

$$\boldsymbol{F}_e \triangleq \boldsymbol{F}_d - \boldsymbol{F} \tag{4.22}$$

where $\boldsymbol{X}_l$ is the position of the leader manipulator used as position reference and $\boldsymbol{F}_d$ is the desired force trajectory. Differentiating (4.21) with respect to time twice gives the velocity and acceleration errors

$$\dot{\boldsymbol{X}}_e = \dot{\boldsymbol{X}}_l - \dot{\boldsymbol{X}} \tag{4.23}$$

$$\ddot{\boldsymbol{X}}_e = \ddot{\boldsymbol{X}}_l - \ddot{\boldsymbol{X}} \tag{4.24}$$

The applied force $\boldsymbol{F}$ is assumed to be unknown; hence its dynamics is not known, and cannot be used in the force control loop. The control signal $\boldsymbol{u}$ in (4.18) will be used to control both position and force by using the selection matrices defined in (4.19) - (4.20):

$$\boldsymbol{u} = \mathbf{S}_p\boldsymbol{\alpha}_p + \mathbf{S}_f\boldsymbol{\alpha}_f \tag{4.25}$$

where $\boldsymbol{\alpha}_p$ is the position controller and $\boldsymbol{\alpha}_f$ is the force controller. $\boldsymbol{\alpha}_p$ will be designed to control the position in all degrees of freedom, but the selection matrix $\mathbf{S}_p$ will nullify the contribution to the force controlled degrees. In a similar fashion, $\boldsymbol{\alpha}_f$ will be designed as a pure force controller of all degrees of freedom, and $\mathbf{S}_f$ will eliminate the irrelevant degrees of freedom.

The position controller $\boldsymbol{\alpha}_p$ is simply designed as a PD controller with feed-forward acceleration:

$$\boldsymbol{\alpha}_p = \ddot{\boldsymbol{X}}_l + \mathbf{K}_{pd}\dot{\boldsymbol{X}}_e + \mathbf{K}_{pp}\boldsymbol{X}_e \tag{4.26}$$

where $\mathbf{K}_{pp}, \mathbf{K}_{pd}$ are diagonal positive definite matrices consisting of position and velocity gains, respectively.

The force controller on the other hand, was not so straightforward to design. First, a term proportional to the force error $\boldsymbol{F}_e$ is needed. This force must be mapped from the constraint frame to the end-effector by using the mobility tensor from (4.17).

At first, let

$$\boldsymbol{\alpha}_f = \mathbf{W}\mathbf{K}_{fp}\boldsymbol{F}_e \tag{4.27}$$

where $\mathbf{K}_{fp}$ is a positive definite diagonal force gain matrix. I suspected that applying the force controller (4.27) to the double integrator system given by (4.18) will give a heavily oscillating system. Therefore, I chose to modify (4.27) by adding a damping term:

$$\boldsymbol{\alpha}_f = \mathbf{W}\mathbf{K}_{fp}\boldsymbol{F}_e - \mathbf{K}_{fd}\dot{\boldsymbol{X}} \tag{4.28}$$

where $\mathbf{K}_{fd}$ is a positive definite diagonal damping gain matrix. The damping term is chosen to be proportional to the actual velocity of the end-effector, not the velocity error. In this way, the damping term will not try to follow a velocity trajectory in the force controlled directions, but will help damping the oscillations.

The final controller is given by

$$\boldsymbol{\tau} = \mathbf{C}\dot{\boldsymbol{q}} + \boldsymbol{g} + \mathbf{M}\mathbf{J}_a^{-1}(\boldsymbol{u} - \dot{\mathbf{J}}_a\dot{\boldsymbol{q}}) + \mathbf{J}^{\mathrm{T}}\boldsymbol{F} \tag{4.29}$$

$$\boldsymbol{u} = \mathbf{S}_p\boldsymbol{\alpha}_p + \mathbf{S}_f\boldsymbol{\alpha}_f \tag{4.30}$$

$$\boldsymbol{\alpha}_p = \ddot{\boldsymbol{X}}_l + \mathbf{K}_{pd}\dot{\boldsymbol{X}}_e + \mathbf{K}_{pp}\boldsymbol{X}_e \tag{4.31}$$

$$\boldsymbol{\alpha}_f = \mathbf{W}\mathbf{K}_{fp}\boldsymbol{F}_e - \mathbf{K}_{fd}\dot{\boldsymbol{X}} \tag{4.32}$$

The stability analysis of the proposed controller given by (4.29) - (4.32) will follow in the next section.

4.4.1 Stability of the position loop

The position controlled and force-controlled degrees of freedom can be extracted from the position vector $\boldsymbol{X}$ using the selection matrices $\mathbf{S}_p$ and $\mathbf{S}_f$ defined in (4.19) - (4.20). Let the filtered state vectors be

$$\boldsymbol{X}^p \triangleq \mathbf{S}_p \boldsymbol{X} \tag{4.33}$$
$$\boldsymbol{X}^f \triangleq \mathbf{S}_f \boldsymbol{X} \tag{4.34}$$

where $\boldsymbol{X}^p \in \mathbb{R}^6$ has zeros in the force-controlled elements, and $\boldsymbol{X}^f \in \mathbb{R}^6$ has zeros in the position controlled elements. Under the assumption that the selection matrices are constant, the time derivatives are given by

$$\dot{\boldsymbol{X}}^p = \mathbf{S}_p \dot{\boldsymbol{X}}, \qquad \ddot{\boldsymbol{X}}^p = \mathbf{S}_p \ddot{\boldsymbol{X}} \tag{4.35}$$
$$\dot{\boldsymbol{X}}^f = \mathbf{S}_f \dot{\boldsymbol{X}}, \qquad \ddot{\boldsymbol{X}}^f = \mathbf{S}_f \ddot{\boldsymbol{X}} \tag{4.36}$$

Following from the definition, the selection matrices has these properties:

$$(\mathbf{S}_p)^n = \mathbf{S}_p \tag{4.37}$$
$$(\mathbf{S}_f)^n = \mathbf{S}_f \tag{4.38}$$
$$\mathbf{S}_p \mathbf{S}_f = \mathbf{S}_f \mathbf{S}_p = \mathbf{0} \tag{4.39}$$

where n is a positive integer. Inserting the controller (4.30) - (4.32) into the system (4.18) gives the closed loop system

$$\begin{aligned} \ddot{\boldsymbol{X}} &= \mathbf{S}_p \boldsymbol{\alpha}_p + \mathbf{S}_f \boldsymbol{\alpha}_f \\ &= \mathbf{S}_p \left[\ddot{\boldsymbol{X}}_l + \mathbf{K}_{pd} \dot{\boldsymbol{X}}_e + \mathbf{K}_{pp} \boldsymbol{X}_e \right] + \mathbf{S}_f \left[\mathbf{W} \mathbf{K}_{fp} \boldsymbol{F}_e - \mathbf{K}_{fd} \dot{\boldsymbol{X}} \right] \end{aligned} \tag{4.40}$$

By pre-multiplying both sides of (4.40) with $\mathbf{S}_p$ and using Equations (4.37) - (4.39), the dynamics of the position controlled degrees of freedom can be expressed as

$$\begin{aligned}\mathbf{S}_p\ddot{\boldsymbol{X}} &= \mathbf{S}_p\mathbf{S}_p\left[\ddot{\boldsymbol{X}}_l + \mathbf{K}_{pd}\dot{\boldsymbol{X}}_e + \mathbf{K}_{pp}\boldsymbol{X}_e\right] + \mathbf{S}_p\mathbf{S}_f\left[\mathbf{W}\mathbf{K}_{fp}\boldsymbol{F}_e - \mathbf{K}_{fd}\dot{\boldsymbol{X}}\right] \\ &= \mathbf{S}_p\left[\ddot{\boldsymbol{X}}_l + \mathbf{K}_{pd}\dot{\boldsymbol{X}}_e + \mathbf{K}_{pp}\boldsymbol{X}_e\right]\end{aligned} \tag{4.41}$$

Because the selection matrices $\mathbf{S}_i$ and the gain matrices $\mathbf{K}_j$ are diagonal, the relation $\mathbf{S}_i\mathbf{K}_j = \mathbf{K}_j\mathbf{S}_i$ can be used. By rearranging (4.41) we get

$$\begin{aligned}\mathbf{S}_p\ddot{\boldsymbol{X}} &= \mathbf{S}_p\ddot{\boldsymbol{X}}_l + \mathbf{S}_p\mathbf{K}_{pd}\dot{\boldsymbol{X}}_e + \mathbf{S}_p\mathbf{K}_{pp}\boldsymbol{X}_e \\ \mathbf{S}_p\ddot{\boldsymbol{X}} &= \mathbf{S}_p\ddot{\boldsymbol{X}}_l + \mathbf{K}_{pd}\mathbf{S}_p\dot{\boldsymbol{X}}_e + \mathbf{K}_{pp}\mathbf{S}_p\boldsymbol{X}_e \\ 0 &= \mathbf{S}_p(\ddot{\boldsymbol{X}}_l - \ddot{\boldsymbol{X}}) + \mathbf{K}_{pd}\mathbf{S}_p\dot{\boldsymbol{X}}_e + \mathbf{K}_{pp}\mathbf{S}_p\boldsymbol{X}_e \\ 0 &= \mathbf{S}_p\ddot{\boldsymbol{X}}_e + \mathbf{K}_{pd}\mathbf{S}_p\dot{\boldsymbol{X}}_e + \mathbf{K}_{pp}\mathbf{S}_p\boldsymbol{X}_e\end{aligned}$$

Finally, by using the definitions (4.33) - (4.36), the dynamics of the position error becomes

$$\ddot{\boldsymbol{X}}_e^p + \mathbf{K}_{pd}\dot{\boldsymbol{X}}_e^p + \mathbf{K}_{pp}\boldsymbol{X}_e^p = 0 \tag{4.42}$$

The stability of the position loop is going to be proven by using Lyapunov theory[4]. Consider the positive definite Lyapunov function candidate

$$V(\boldsymbol{X}_e^p, \dot{\boldsymbol{X}}_e^p) = \frac{1}{2}(\dot{\boldsymbol{X}}_e^p)^{\mathrm{T}}\dot{\boldsymbol{X}}_e^p + \frac{1}{2}(\boldsymbol{X}_e^p)^{\mathrm{T}}\mathbf{K}_{pp}\boldsymbol{X}_e^p \tag{4.43}$$

[4]For more on Lyapunov theory and stability analysis, see Khalil [4].

The time derivative of (4.43) along the error dynamics given by (4.42) is

$$\begin{aligned}
\dot{V} &= (\dot{\boldsymbol{X}}_e^p)^{\mathrm{T}} \ddot{\boldsymbol{X}}_e^p + (\dot{\boldsymbol{X}}_e^p)^{\mathrm{T}} \mathbf{K}_{pp} \boldsymbol{X}_e^p \\
&= (\dot{\boldsymbol{X}}_e^p)^{\mathrm{T}} \left[\mathbf{K}_{pp} \boldsymbol{X}_e^p + \ddot{\boldsymbol{X}}_e^p \right] \\
&= (\dot{\boldsymbol{X}}_e^p)^{\mathrm{T}} \left[\mathbf{K}_{pp} \boldsymbol{X}_e^p - \mathbf{K}_{pd} \dot{\boldsymbol{X}}_e^p - \mathbf{K}_{pp} \boldsymbol{X}_e^p \right] \\
\dot{V} &= -(\dot{\boldsymbol{X}}_e^p)^{\mathrm{T}} \mathbf{K}_{pd} \dot{\boldsymbol{X}}_e^p
\end{aligned} \tag{4.44}$$

which is only negative semi-definite. This proves stability, but not asymptotically stability. By applying LaSalle's invariance principle, we get

$$\begin{aligned}
\dot{V} \equiv 0 \;\Rightarrow\; \dot{\boldsymbol{X}}_e^p \equiv \mathbf{0} \;\Rightarrow\; \ddot{\boldsymbol{X}}_e^p \equiv \mathbf{0} \\
\Rightarrow\; \mathbf{K}_{pp} \boldsymbol{X}_e^p \equiv \mathbf{0} \;\Rightarrow\; \boldsymbol{X}_e^p \equiv \mathbf{0}
\end{aligned}$$

This implies that the only invariant set satisfying $\dot{V} = 0$ is the origin. Hence, the position control loop is asymptotically stable. Furthermore, since the Lyapunov function V given by (4.43) is radially unbounded, this result is global. Thus, the position loop is proven to be globally asymptotically stable.

Note that this is only valid as long as the analytical Jacobian $\mathbf{J}_a$ has full rank.

4.4.2 Stability of the force loop

Then it remains to analyze the stability of the force controlled degrees of freedom. Inserting (4.30) - (4.32) into (4.18) and pre-multiplying both sides with the selection matrix $\mathbf{S}_f$ to extract the force controlled degrees of freedom gives

$$\begin{aligned}\mathbf{S}_f\ddot{\boldsymbol{X}} &= \mathbf{S}_f\mathbf{S}_p\left[\ddot{\boldsymbol{X}}_l + \mathbf{K}_{pd}\dot{\boldsymbol{X}}_e + \mathbf{K}_{pp}\boldsymbol{X}_e\right] + \mathbf{S}_f\mathbf{S}_f\left[\mathbf{W}\mathbf{K}_{fp}\boldsymbol{F}_e - \mathbf{K}_{fd}\dot{\boldsymbol{X}}\right] \\ &= \mathbf{S}_f\left[\mathbf{W}\mathbf{K}_{fp}\boldsymbol{F}_e - \mathbf{K}_{fd}\dot{\boldsymbol{X}}\right]\end{aligned}$$

Rearranging according to the previous method gives

$$\begin{aligned}\mathbf{S}_f\ddot{\boldsymbol{X}} &= \mathbf{S}_f\mathbf{W}\mathbf{K}_{fp}\boldsymbol{F}_e - \mathbf{S}_f\mathbf{K}_{fd}\dot{\boldsymbol{X}} \\ &= \mathbf{S}_f\mathbf{W}\mathbf{K}_{fp}\boldsymbol{F}_e - \mathbf{K}_{fd}\mathbf{S}_f\dot{\boldsymbol{X}} \\ \ddot{\boldsymbol{X}}^f &= \mathbf{S}_f\mathbf{W}\mathbf{K}_{fp}\boldsymbol{F}_e - \mathbf{K}_{fd}\dot{\boldsymbol{X}}^f \\ \ddot{\boldsymbol{X}}^f + \mathbf{K}_{fd}\dot{\boldsymbol{X}}^f &= \mathbf{S}_f\mathbf{W}\mathbf{K}_{fp}\boldsymbol{F}_e \qquad (4.45)\end{aligned}$$

It is assumed that the mobility tensor $\mathbf{W}$ is bounded, and if the force error $\boldsymbol{F}_e$ is bounded, the system described by (4.45) will behave as a mass-damper system with identity mass and damper coefficient $\mathbf{K}_{fd}$, driven by a force $\mathbf{S}_f\mathbf{W}\mathbf{K}_{fp}\boldsymbol{F}_e$. By choosing $\mathbf{K}_{fd}$ sufficiently large, the oscillations of the system will be damped satisfactorily.

The dynamics of the force error $\boldsymbol{F}_e$ is not known, but is related to the position of the end effector when it is in contact with the environment. The position of the end-effector normal to the surface will be driven towards the point where the force applied is equal to the desired force, but it is not certain that this point will be reached. Hence, the force loop appears to be stable, and this will be verified by simulations.

4.4.3 Reduction to pure position controller

It is favorable that the controller given by (4.29) - (4.30) reduces to a pure position controller when appropriate. Then the robot manipulator avoid the need for another control concept to handle tasks not including environmental contact.

When the end-effector is not in contact with the environment, the external force F exerted is zero, and the equations of motion for the manipulator reduces to

$$\mathbf{M}\ddot{\boldsymbol{q}} + \mathbf{C}\dot{\boldsymbol{q}} + \boldsymbol{g} = \boldsymbol{\tau}$$

By choosing the selection matrices as $\mathbf{S}_p = \mathbf{I}$ and $\mathbf{S}_f = \mathbf{0}$ to select pure position control, the controller reduces to

$$\begin{aligned}\boldsymbol{\tau} &= \mathbf{C}\dot{\boldsymbol{q}} + \boldsymbol{g} + \mathbf{M}\mathbf{J}_a^{-1}(\boldsymbol{u} - \dot{\mathbf{J}}_a\dot{\boldsymbol{q}}) \\ \boldsymbol{u} &= \ddot{\boldsymbol{X}}_l + \mathbf{K}_{pd}\dot{\boldsymbol{X}}_e + \mathbf{K}_{pp}\boldsymbol{X}_e\end{aligned}$$

Insertion gives the closed loop system

$$\ddot{\boldsymbol{X}}_e + \mathbf{K}_{pd}\dot{\boldsymbol{X}}_e + \mathbf{K}_{pp}\boldsymbol{X}_e = \mathbf{0}$$

which is globally asymptotically stable for positive definite diagonal matrices $\mathbf{K}_{pp}, \mathbf{K}_{pd}$.

4.5 Comparison between the control concepts

This section will compare the stability properties and practical issues regarding the three hybrid control concept presented in this thesis: The hybrid force/motion concept presented by Raibert and Craig [9], the dynamic hybrid control scheme by Yoshikawa [13], and the new hybrid controller proposed in this master's thesis.

The hybrid control concept of Raibert and Craig introduced an easy and straightforward way to separate the degrees of freedom in position controlled and force controlled ones. By transforming the coordinates of the end-effector to a conveniently placed constraint frame, the separation is merely a multiplication with a constant selection matrix. On the other hand, Raibert and Craig did not compensate for the manipulator dynamics. Therefore, it has not been proven that both the position and force trajectories can be tracked simultaneously; the authors showed the concept's performance by simulations and experiments. Raibert and Craig did also control in joint space, requiring the use of inverse kinematics, or the inverse Jacobian as an approximation.

Yoshikawa defined the constraints on the end-effector differently, by describing them as hypersurfaces. These constraints gave the basis for non-constant selection matrices, and the constraint frame is not necessarily orthogonal. His method was not as intuitive and simple as the Raibert and Craig concept, but he did prove stability for both the position and force loop.

The hybrid control concept proposed in this master's thesis retains the simple and intuitive separation technique used by Raibert and Craig. Contrary to their approach, this method involves compensation of the manipulator dynamics and control in task space. This eludes the need for inverse kinematics solutions. The proposed hybrid control concept was proven to be globally asymptotically stable in the position controlled degrees of freedom, and the stability of the force loop was argued to be stable.

Chapter 5

Simulations

After the analysis of the controller derived in the previous chapter, it is desirable to verify the results by simulation. This will illustrate the performance of the control concept, and its implications for practical use.

A model of a rigid robot manipulator with six revolute joints was implemented in MATLAB/Simulink. This software package was chosen because I am quite acquainted with it, and it is designed for dynamic models and matrix operations.

The kinematic structure of the robot shown in Figure 5.1 is similar to the *KUKA KR16* manipulator at the SINTEF robotic lab facility, and is given by Table A.1. The physical parameters of the lab robot, such as link masses and moments of inertia, are currently unknown; thus the simulations are done with artificial values. These values are listed in Table A.2. The numeric integration method selected was Runge-Kutta 4. order (RK4), and the time step was chosen to be 0.01 s.

In these simulations, the task of the robot manipulator was to track both a position trajectory along a wall and a force trajectory normal to the wall. For example, this can be a robot washing a window or polishing a surface, with the end-effector orientation normal to the surface. Control of the force exerted on the environment is essential to be able to remain in contact with the surface and prevent damage.

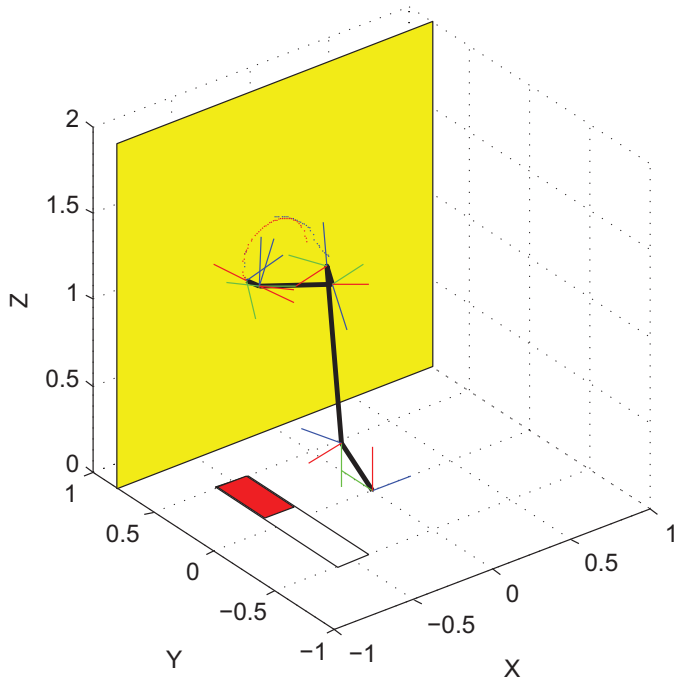

Figure 5.1: A snapshot of the robot simulator. The meter on the floor shows the applied force against the wall.

In the simulations, the normal Euler angle equations were used, introducing representational singularities at $\theta = \pm\pi$. As mentioned in the previous chapter, this can be avoided using alternative representations, but for simplicity I chose to not implement such a scheme and simulate in a space where the effects of these singularities were minimal.

To suppress the effects of kinematic singularities in the geometric Jacobian, I used a damped pseudoinverse Jacobian matrix given by

$$\mathbf{J}_d^{\dagger} = \mathbf{J}^{\mathrm{T}}(\mathbf{J}\mathbf{J}^{\mathrm{T}} + \gamma\mathbf{I})^{-1}$$

where γ is a small positive constant giving the damping ratio. The parameters of the position PD controller were chosen to make the system critically damped. All the controller parameters are given in Table A.3.

In all the following simulations, the position of the wall was $y = 0.8\,\mathrm{m}$, and it is assumed that forces caused by interaction with the environment is only working in the y-direction.

The corresponding selection matrices was

$$\mathbf{S}_p = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{S}_f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The initial value of the end-effector position was set to

$$\boldsymbol{X}(0) = \begin{bmatrix} 0 & 0.6 & 1 & 0 & 1 & 0 \end{bmatrix}^{\mathrm{T}} \tag{5.1}$$

to make the end-effector start some distance from the wall, with an error in orientation. The initial velocity was set to zero.

5.1 Force applied against the environment

When simulating the interaction between the end-effector and the environment, some method to determine the exerted force in the simulations is needed to emulate a wrist mounted force sensor. This is done by reading the position of the end-effector close to the wall, and define a function mapping a small position change to a force.

One option is to model the wall as a stiff spring. By measuring the position deviation from the nominal wall position, one can determine the applied force. However, as I wanted the environment to appear as rigid, and not as a spring, my choice of force model landed on a repulsive force defined in Spong et al. [10].

The force acting in one degree of freedom is

$$\boldsymbol{F} = \begin{cases} \eta \left(\frac{1}{\rho} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2} & , \rho \leq \rho_0 \\ 0 & , \rho > \rho_0 \end{cases}$$

where ρ is the distance to the object, and ρ_0 is the distance of influence; i.e. the area close to the environment where the force is active. At a distance greater than ρ_0 the repulsive force is set to zero to avoid interfering with the movement of the manipulator in free space.

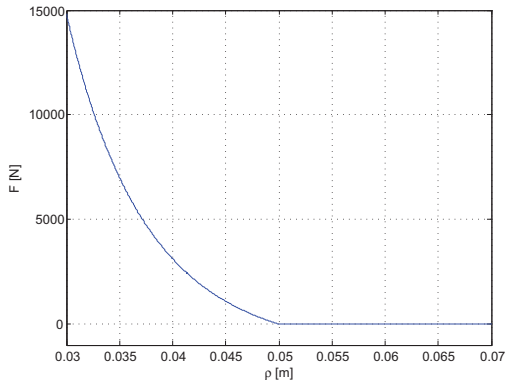

Figure 5.2: The repulsive force. $\rho_0 = 0.05\,\mathrm{m}$

5.2 Constant position reference

To begin with, the system was simulated with a constant position reference given by

$$\boldsymbol{X}_l = \begin{bmatrix} 0 & 0.8 & 1 & 0 & 0 & -\pi \end{bmatrix}^{\mathrm{T}} \tag{5.2}$$

where $y = 0.8\,\mathrm{m}$ is on the wall and $\Theta = \begin{bmatrix} 0 & 0 & -\pi \end{bmatrix}^{\mathrm{T}}$ makes the end-effector orientation normal to the wall. Several time-varying force references where tried, and will be presented below. The desired force F_d is a scalar function describing the force trajectory in the y-direction.

5.2.1 Step force reference

Firstly, a constant force reference with a step was applied. Then the system's response to both a constant force and a step will be investigated. The desired force F_d was given by

$$F_d = 10 + 10\,u(t-5) \tag{5.3}$$

where $u(t-\tau)$ is the Heaviside step function initiating a unit step at $t = \tau\,\mathrm{s}$.

The results from this simulation can be seen in Figure 5.3.

5.2.2 Ramp force reference

Secondly, the force reference F_d was chosen to be a ramp given by

$$F_d = 10 + t \tag{5.4}$$

The results from this simulation can be seen in Figure 5.4.

5.2.3 Sine force reference

Thirdly, the desired force F_d was chosen to be

$$F_d = 10 + 2\sin(2t) \tag{5.5}$$

The results from this simulation can be seen in Figure 5.5.

5.2.4 Discussion

As can be seen from Figures 5.3 - 5.5, the proposed hybrid controller is able to make the system converge to the desired force and position trajectory when the position is constant.

The shape of the force trajectory has little effect on the force error and position error. The velocity and force errors in Figure 5.3, on the other hand, is experiencing a spike at $t = 5\,\mathrm{s}$. This is caused by the step in the force reference.

The end-effector's initial position in these simulations is some distance from the wall. The controller will drive the manipulator against the wall to achieve the desired force. All three simulations show a spike in the force at $t \approx 2\,\mathrm{s}$. This is caused by the impact when the end-effector collides the wall, and may cause problems if it is important to restrict the exerted force on the environment. This problem will not be addressed in this thesis.

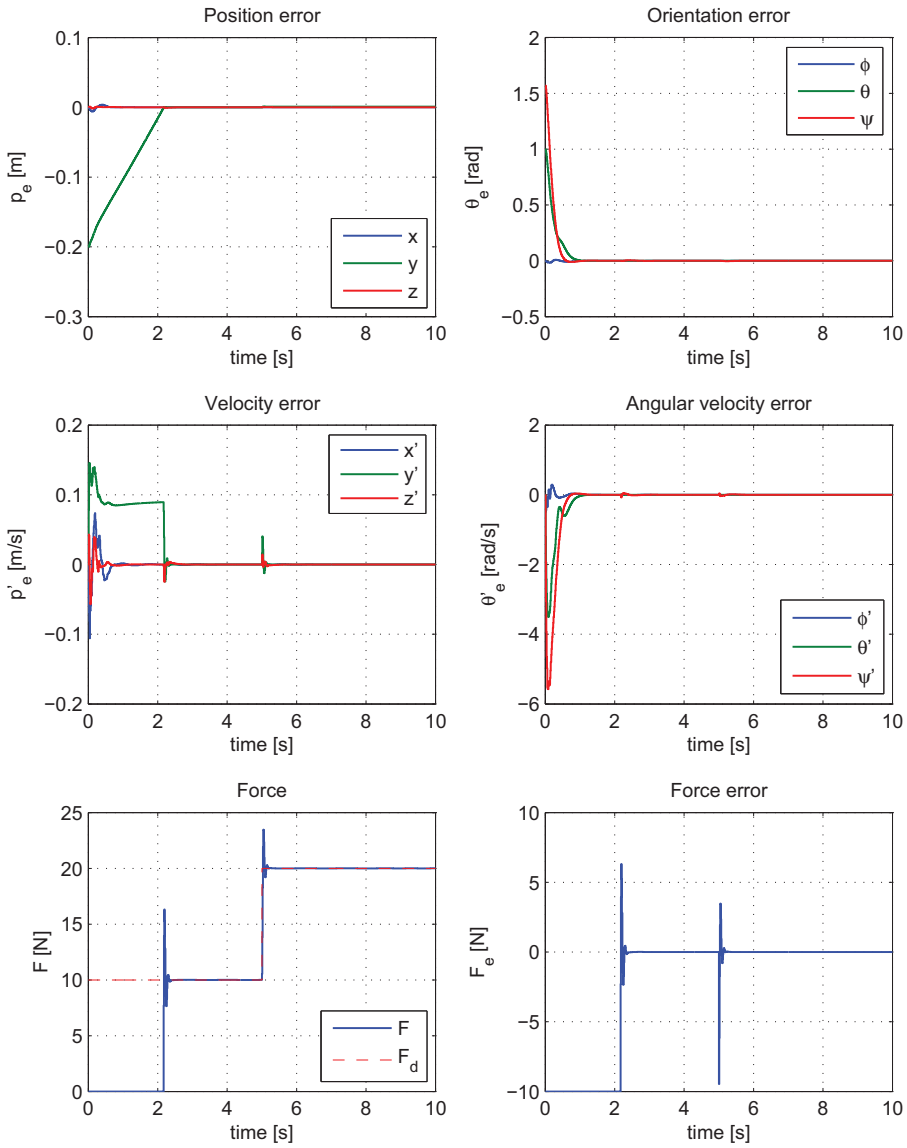

Figure 5.3: Simulation: Constant position reference, step force reference.

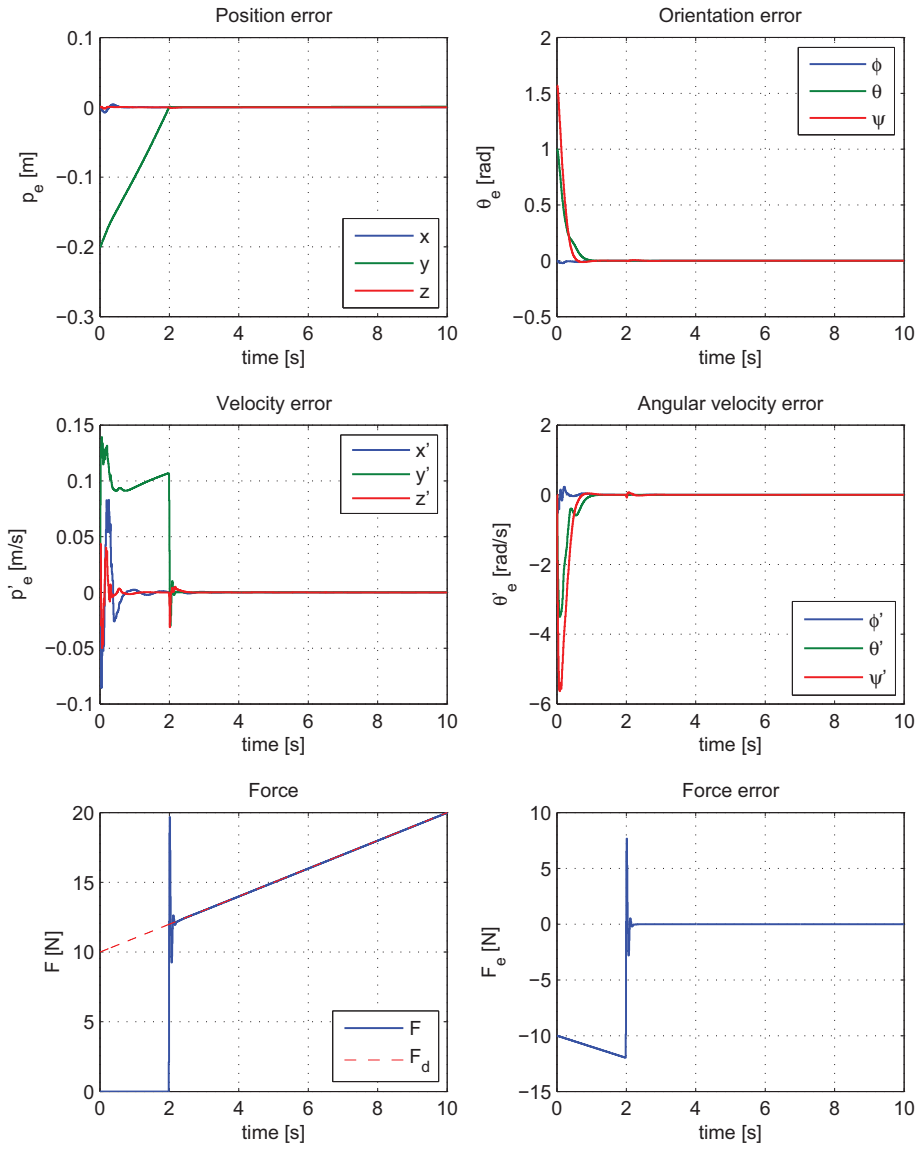

Figure 5.4: Simulation: Constant position reference, ramp force reference.

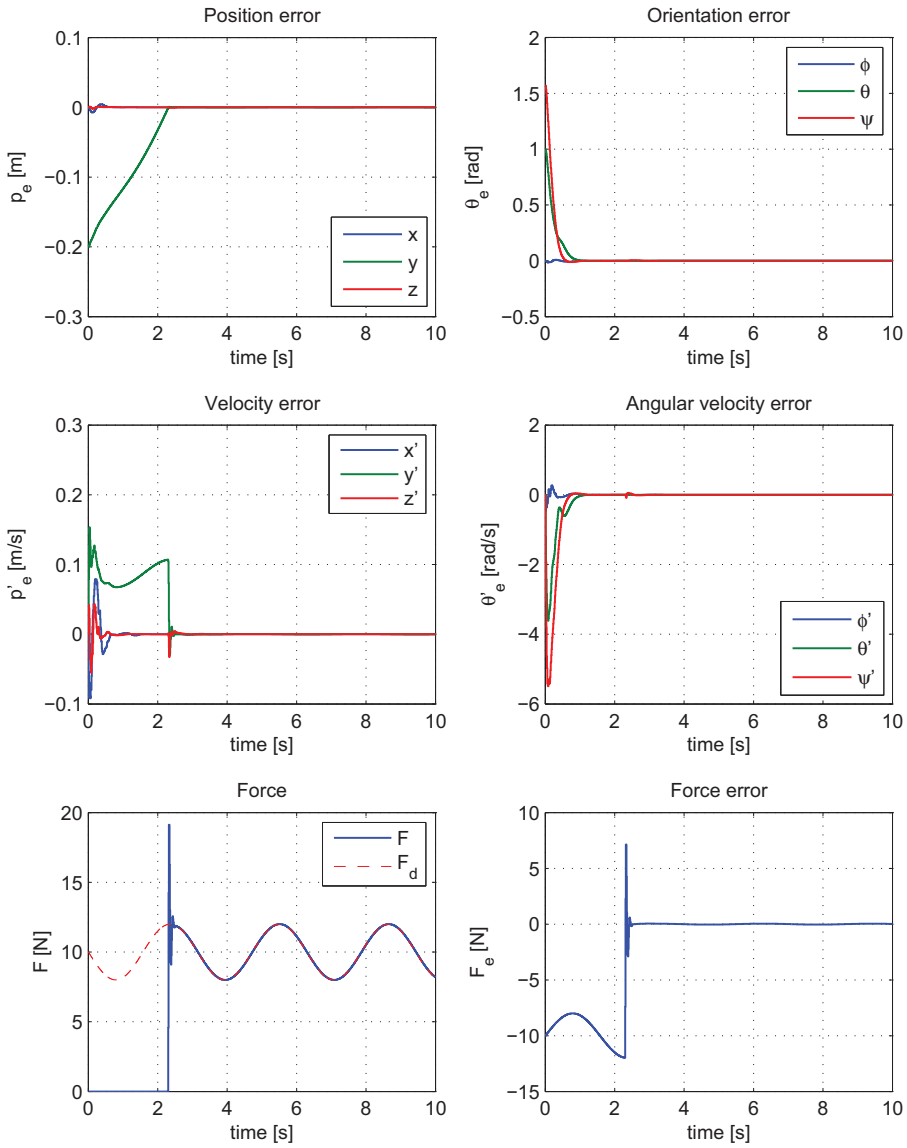

Figure 5.5: Simulation: Constant position reference, sine force reference.

5.3 Varying position reference

After seeing how the system behaves under a constant position reference, it is time to try a time-varying position trajectory. The task was to trace a circle on the wall. The radius of the circle was set to be 0.2 m, and its center was located at $\begin{bmatrix}0 & 0.8 & 1\end{bmatrix}^{\mathrm{T}}$. The reference orientation of the end-effector was still normal to the wall, and the circle would be traced with an angular velocity of 0.9 rad/s.

The reference position, velocity and acceleration was given by

$$\boldsymbol{X}_l = \begin{bmatrix} 0.2\cos(0.9t) \\ 0.8 \\ 1 + 0.2\sin(0.9t) \\ 0 \\ 0 \\ -\pi \end{bmatrix} \tag{5.6}$$

$$\dot{\boldsymbol{X}}_l = \begin{bmatrix} -0.18\sin(0.9t) \\ 0 \\ 0.18\cos(0.9t) \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.7}$$

$$\ddot{\boldsymbol{X}}_l = \begin{bmatrix} -0.162\cos(0.9t) \\ 0 \\ -0.162\sin(0.9t) \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.8}$$

The force references given by (5.3) - (5.5) was used again to make the simulations comparable. The results from these three simulations are illustrated in Figures 5.6 - 5.8, respectively.

5.3.1 Discussion

When comparing the simulations with constant position reference, shown in Figures 5.3 - 5.5, with the simulations with time-varying position reference, given by Figures 5.6 - 5.8, we see that the controller is still able to converge to the desired position and velocity trajectories. Note that both the velocity and force errors are more "bumpy" than in the stationary case.

The force does not quite reach the desired value, but is close enough for practical purposes. This error can possibly be lowered by increasing the force gain $\mathbf{K}_{fp}$.

The form of force reference F_d still has little impact on the errors, except for the spike caused by the step in Figure 5.6.

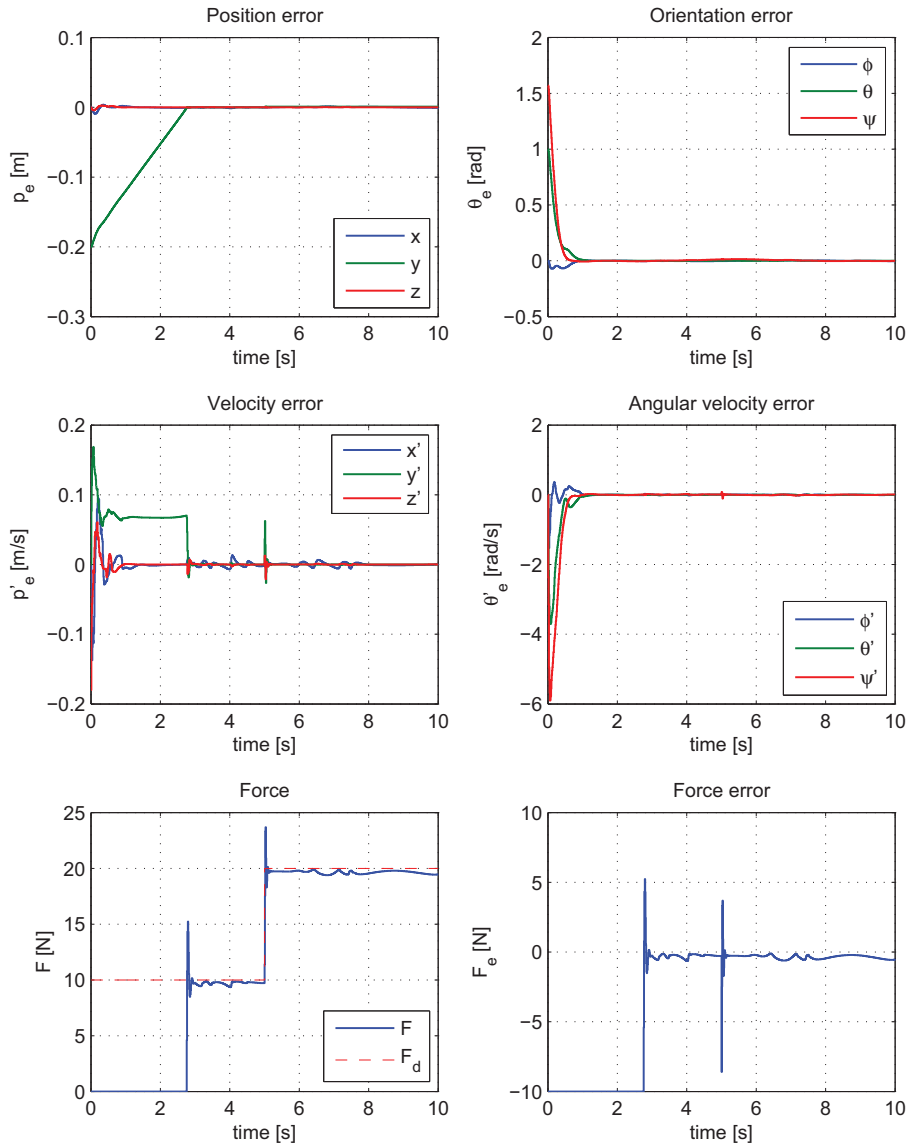

Figure 5.6: Simulation: Varying position reference, step force reference.

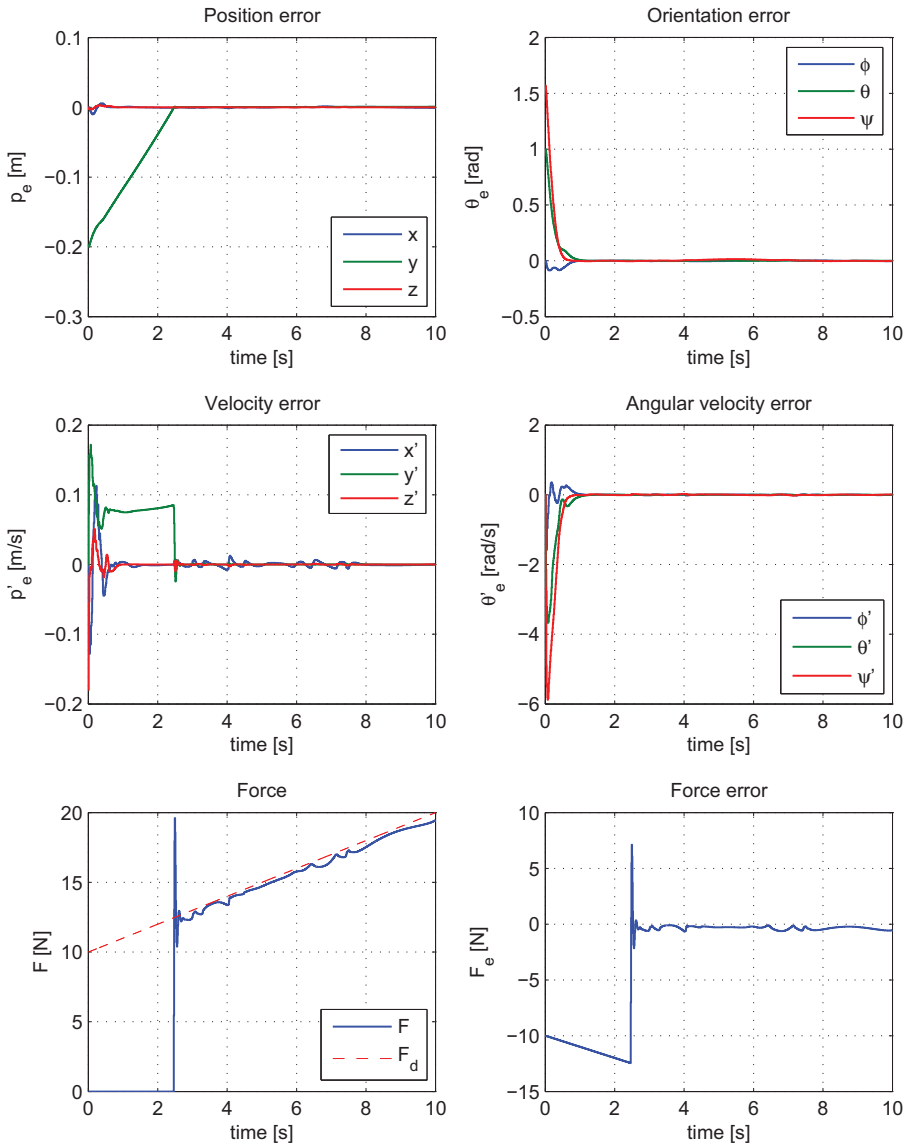

Figure 5.7: Simulation: Varying position reference, ramp force reference.

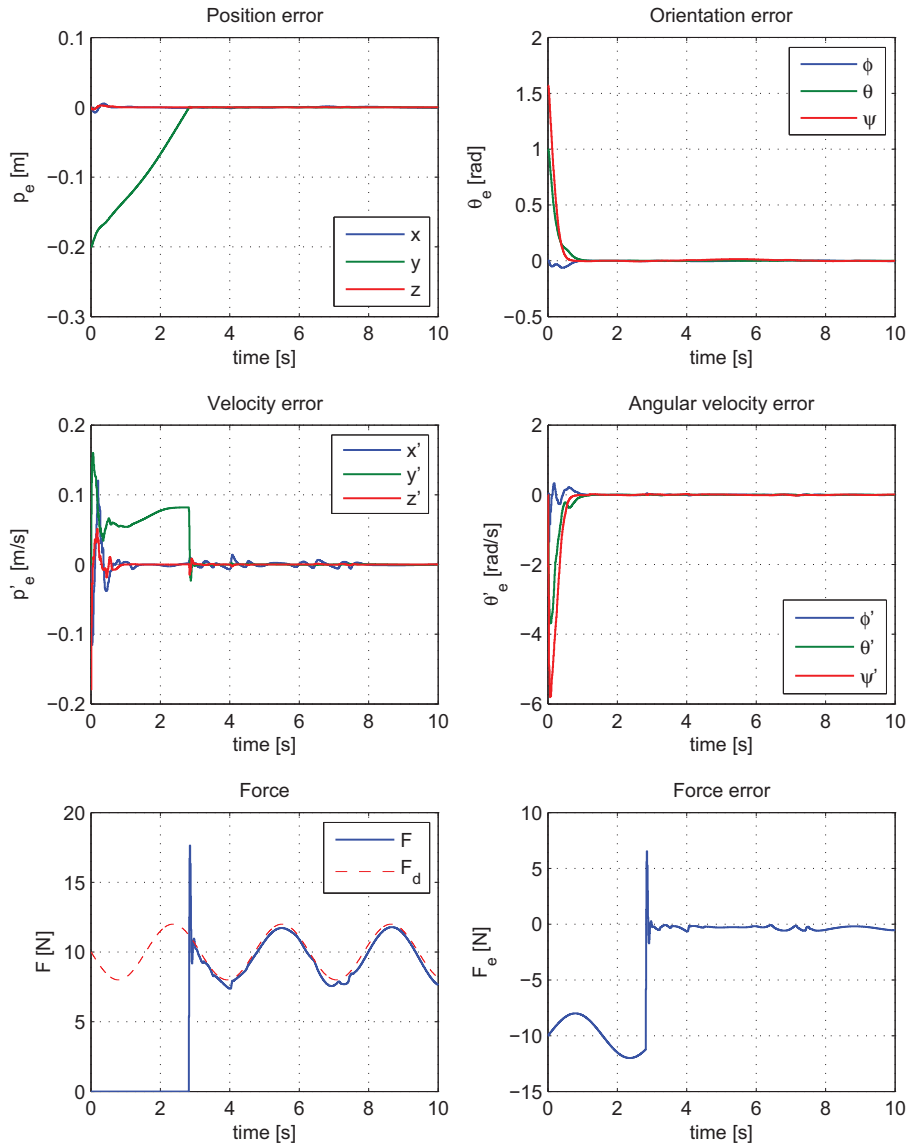

Figure 5.8: Simulation: Varying position reference, sine force reference.

5.4 Zero force reference

Now that the performance of the proposed hybrid control concept has been shown, it is interesting to find out what will happen if the desired force F_d in the force controlled degree of freedom is set to zero; in these simulations the y-direction. This degree of freedom will not be position controlled, and the measured force is zero whenever the manipulator is not touching the wall. Then the controller loses control of this degree of freedom.

The results from this simulation can be seen in Figures 5.9 - 5.10.

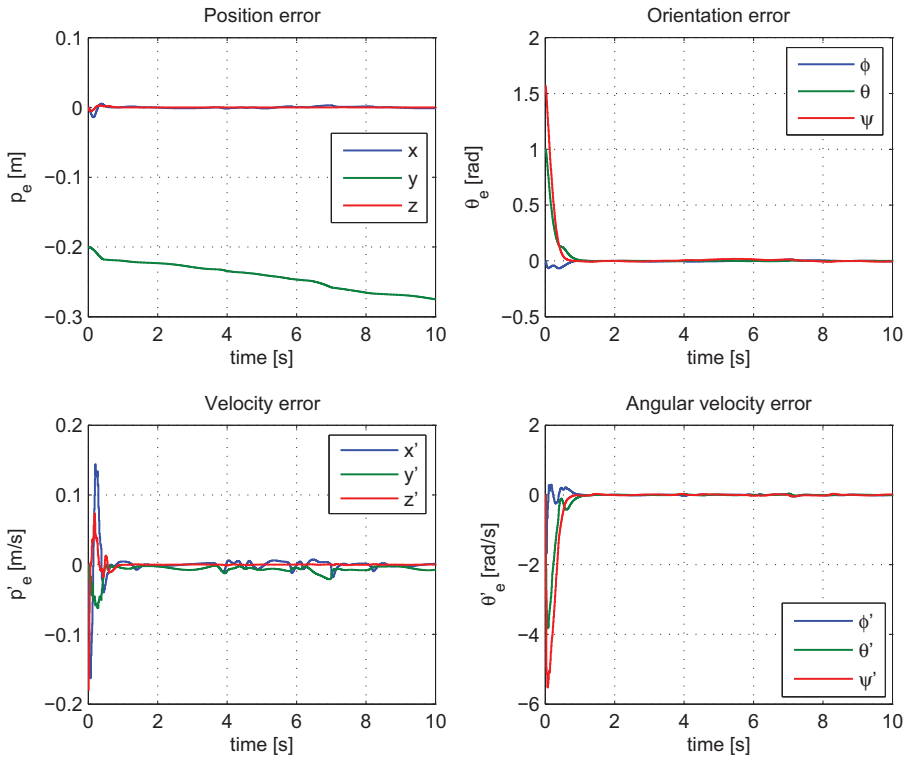

Figure 5.9: Simulation: Varying position reference, zero force reference.

As suspected, the position in the force controlled degree of freedom, y, is under no control and will drift away from the surface, failing to ever achieve contact. This simulations emphasizes the importance of specifying non-zero desired forces in the force controlled degrees of freedom to achieve and maintain contact.

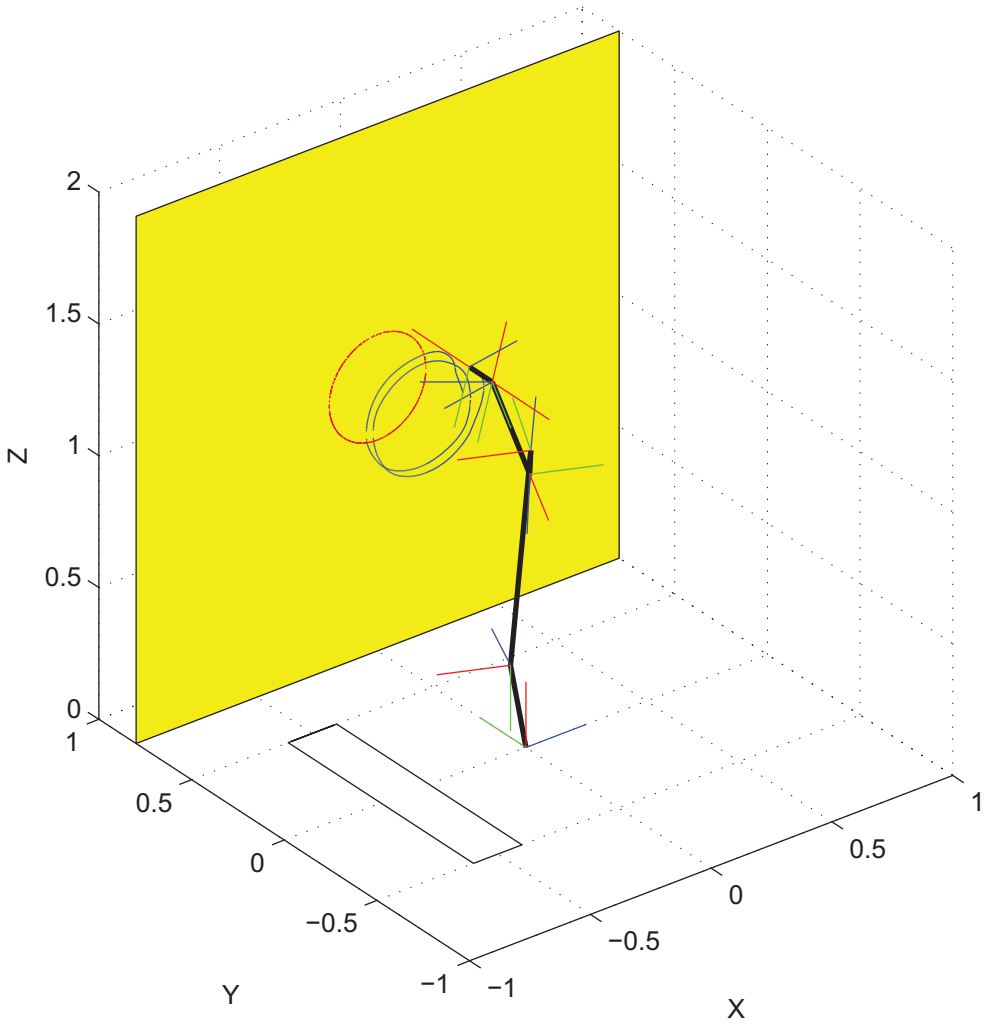

Figure 5.10: XYZ plot - zero force reference.

5.5 Pure position control

By choosing the selection matrices as

$$\mathbf{S}_p = \mathbf{I}, \qquad \mathbf{S}_f = \mathbf{0}$$

pure position control is selected. Then the controller will try to track the position trajectory in all degrees of freedom, and will not take force measurements into consideration.

In the following simulations, the end-effector is still interacting with the wall, and will be influenced by the force exerted, but the controller will ignore the force measurements. The results from the simulation with exactly known environment can be seen in Figure 5.11. The exerted force is within an acceptable range, but there is no direct control of it.

Then the system was simulated with a small difference between the reference trajectory and the actual environment. Figure 5.12 shows the results when the position controller believed the wall was 0.5% closer than it actually was. The result was that the end-effector never achieved contact with the surface.

In the final simulation the reference trajectory was set 0.5% further away from the robot manipulator than the wall actually was. The result, as shown by Figure 5.13, was that in pursuing the position trajectory, the end-effector applied an excessive force against the wall.

Selecting pure position control makes the controller disregard force measurements, and tries to stay in contact with the wall based on position/velocity measurements only. These simulations emphasize the importance of force control when interacting with the environment. The hybrid controller will reach the desired force because it disregards the position trajectory in the force controlled degrees of freedom.

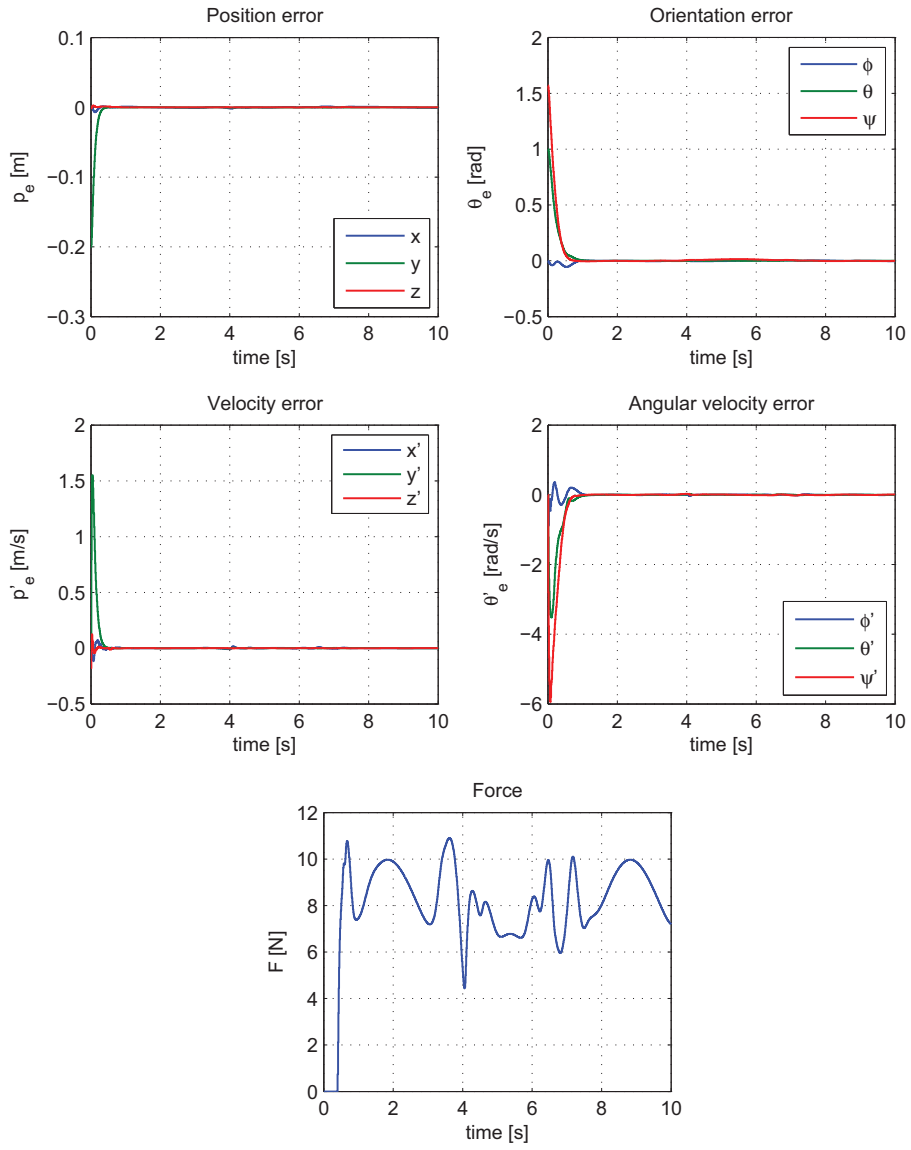

Figure 5.11: Simulation: Pure position control.

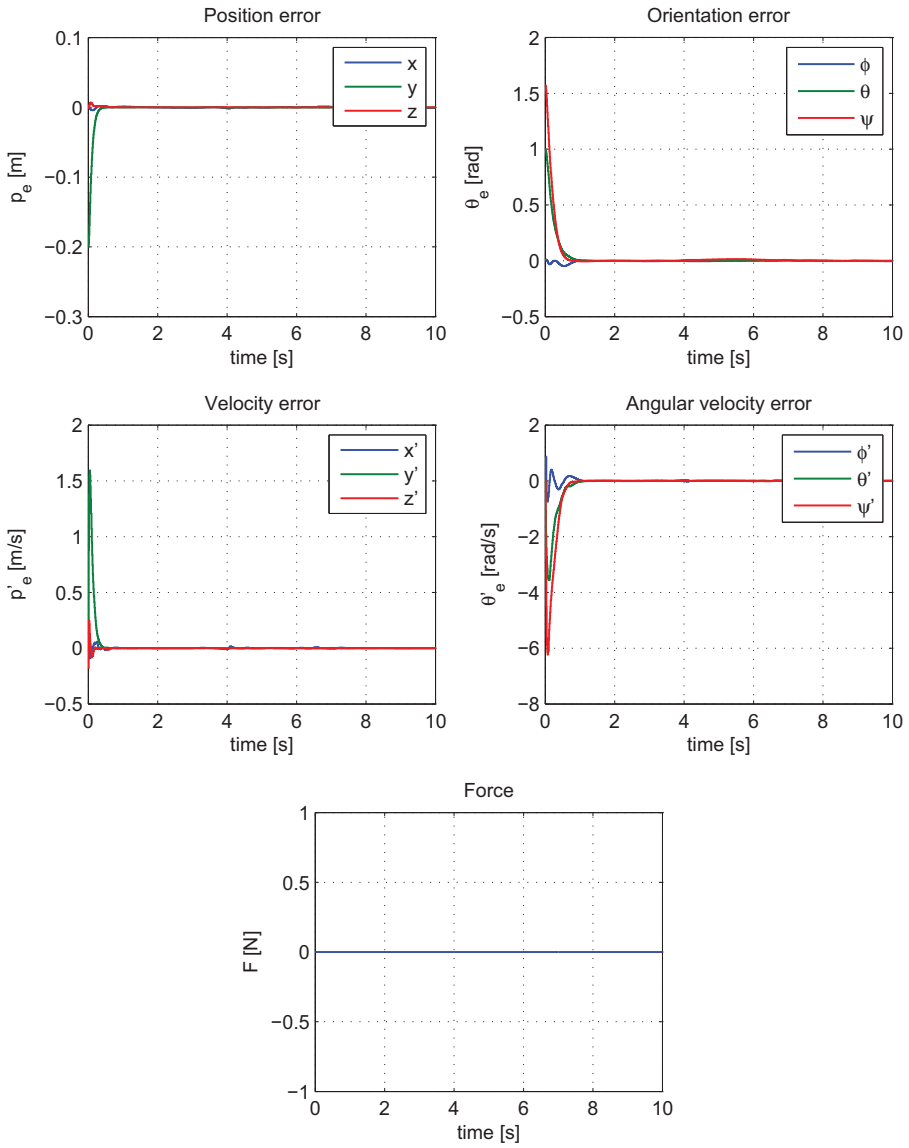

Figure 5.12: Simulation: Pure position control with loss of contact.

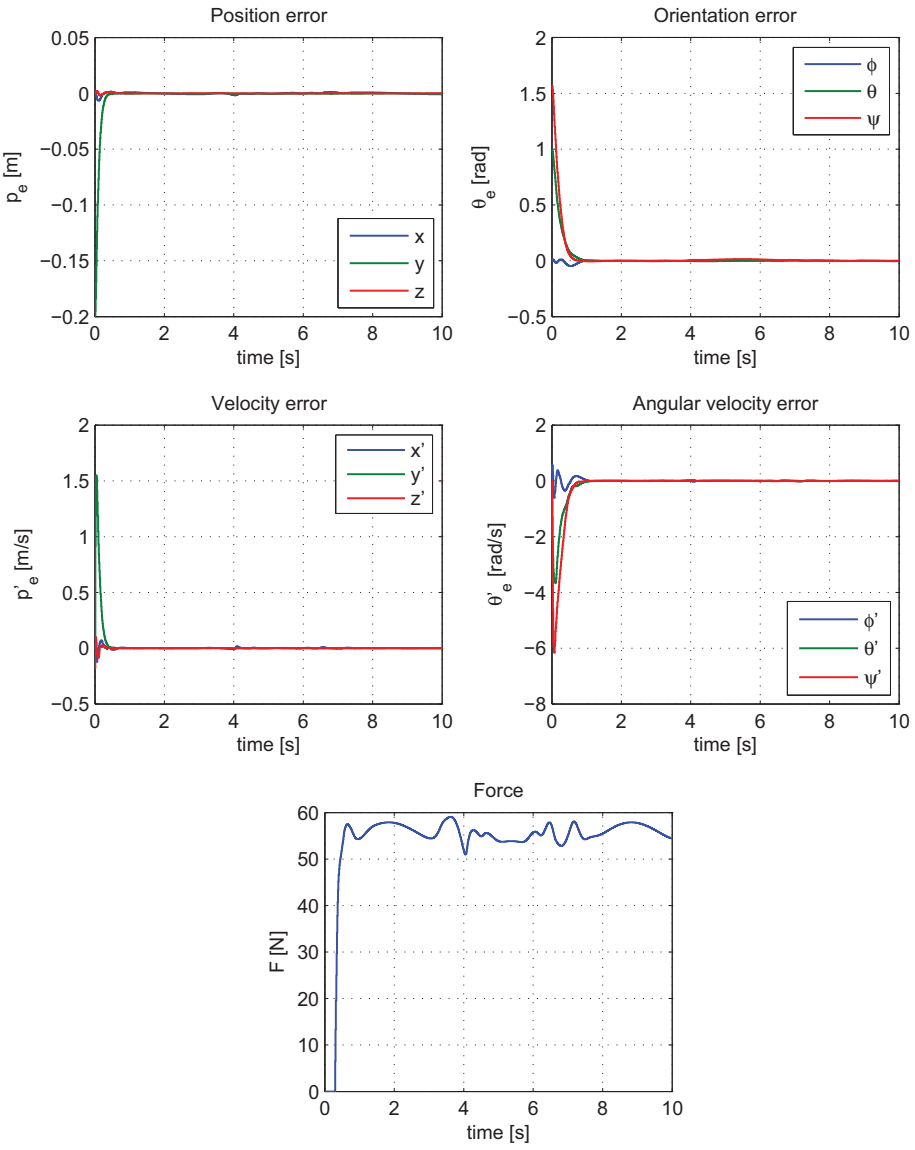

Figure 5.13: Simulation: Pure position control with excessive force.

5.6 Final thoughts

The simulations presented in this chapter verifies the analysis on the proposed hybrid control concept. However, some aspects regarding implementation and practical use remains.

In this thesis limitless input torque on the joint angles was assumed. This is not realistic, and simulations with limited actuators should be performed. Though, more realistic robot model parameters must be provided.

The constraints was assumed to be constant, and it is not known how changing task configurations will affect the proposed control scheme. This should be addressed in the future.

The proposed hybrid control concept is computationally heavy. Quite a few matrices and vectors must be evaluated at each time step to compensate for the robot manipulator dynamics. If this controller is going to be implemented on a micro controller, some form of optimization is needed.

Chapter 6

Conclusion and Further Work

6.1 Conclusion

This master's thesis has resulted in a hybrid force/motion controller for a robot manipulator in a coordinated synchronization scheme. The control problem is separated into two subproblems: position control and force control.

The closed loop system is shown to be globally asymptotically stable in position and velocity in the position controlled degrees of freedom. The system is also argued to be bounded in the force controlled degrees of freedom, but this is not proven yet. These results where verified by simulations.

One of the benefits with this hybrid control concept is that the desired trajectory in the force controlled degrees of freedom is not treated as a disturbance; the errors in these directions are simply ignored by the controller. Hence, if the working environment is quite inaccurate, the difference between the calculated and real surface in the normal direction is ignored by the position controller; the force applied by the end effector is taken care of by the force controller.

When pure position control is needed, the correct selection matrices will reduce the proposed hybrid force/motion controller to a position controller.

6.2 Further work

When the task configuration changes, the selection matrices need modification. In this master's thesis these matrices where assumed to be constant, and the case with time-varying selection matrices must be examined in the future.

The discrete switching from "moving through free space" to "in contact with environment" causes some difficulties for the proposed hybrid controller. This resulted in spikes in the applied force during simulations, and must be addressed.

The stability of the force loop must be further investigated. Though the simulations indicate stability, it is desirable to prove stability theoretically, for example using Lyapunov theory [10].

To avoid the effect of representational singularities, the implementation of a singularity-free orientation representation is needed. To maintain a minimal orientation representation with better singularity handling, consider the approach of [1].

In this thesis it was assumed that all measurements was available, both from the leader and follower manipulator. The dynamic models were also exactly known. In a real life scenario this is unfeasible, and velocities and accelerations should be estimated using observers, for example nonlinear model based observers [5] [7]. The effects modeling errors have on stability should be examined [2]. To compensate for parametric uncertainty, robust or adaptive control concepts can be applied [10].

It would be beneficial to perform experiments at the SINTEF robotic lab facility to verify the analysis and simulations. However, the dynamic model of the robot is currently unknown.

Appendices

Appendix A

Parameters

A.1 Robot parameters

Table A.1: DH parameters

Link	θ_i	d	a	α
0	$\frac{\pi}{2}$	0	0	π
1	q_1	-0.2	0.2	$\frac{\pi}{2}$
2	q_2	0	1	0
3	$\frac{\pi}{2} + q_3$	0	0.1	$-\frac{\pi}{2}$
4	q_4	-0.5	0	$\frac{\pi}{2}$
5	q_5	0	0	$-\frac{\pi}{2}$
6	$\pi + q_6$	-0.1	0	π

Table A.2: Mass and inertia parameters

Link	m_i [kg]	$\mathbf{I}_i$ [kgm^2]	lc_i [m]
1	10	diag([1, 1, 1])	0.1
2	10	diag([1, 1, 1])	0.5
3	10	diag([1, 1, 1])	0.3
4	1	diag([0.1, 0.1, 0.1])	0.1
5	1	diag([0.1, 0.1, 0.1])	0.025
6	1	diag([0.1, 0.1, 0.1])	0.075

where m_i is the link mass, $\mathbf{I}_i$ is the link moment of inertia and lc_i is the distance to the center of gravity of the link.

A.2 Controller parameters

Table A.3: Controller parameters

Gain matrix	Value
$\mathbf{K}_{pp}$	$\text{diag}([400, 400, 400, 100, 100, 100])$
$\mathbf{K}_{pd}$	$\text{diag}([40, 40, 40, 20, 20, 20])$
$\mathbf{K}_{fp}$	$\text{diag}([10, 10, 10, 1, 1, 1])$
$\mathbf{K}_{fd}$	$\text{diag}([40, 40, 40, 10, 10, 10])$

The choice of K_{pp}, K_{pd} makes the position controller critically damped.

References

[1] F. Caccavale, C. Natale, B. Siciliano, and L. Villani. Resolved-acceleration control of robot manipulators: A critical review with experiments. *Robotica*, 16(05):565–573, 1998.

[2] C.C. Cheah, S. Kawamura, and S. Arimoto. Stability of hybrid position and force control for robotic manipulator with kinematics and dynamics uncertainties. *Automatica*, 39(5):847–855, 2003.

[3] O. Egeland and J.T. Gravdahl. *Modeling and simulation for automatic control*. Marine Cybernetics, 2002.

[4] H.K. Khalil. *Nonlinear systems*. Prentice Hall, 2002.

[5] E. Kyrkjebø. *Motion Coordination of Mechanical Systems: Leader-Follower Synchronization of Euler-Lagrange Systems Using Output Feedback Control*. PhD thesis, Norwegian University of Science and Technology, Faculty of Information Technology, Mathematics and Electrical Engineering, 2007.

[6] M.T. Mason. Compliance and Force Control for Computer Controlled Manipulators. *IEEE Trans. Sys. Man Cyb*, 6:418–432, 1981.

[7] H. Nijmeijer and A. Rodriguez-Angeles. *Synchronization of mechanical systems*. World Scientific Pub Co Inc, 2003.

[8] M. Ohto and H. Mayeda. A hybrid position/force control for robot manipulators withposition controllers. In *Industrial Electronics, Control and Instrumentation, 1991. Proceedings. IECON'91., 1991 International Conference on*, pages 1037–1042, 1991.

[9] M.H. Raibert and J.J. Craig. Hybrid position/force control of manipulators. In *American Society of Mechanical Engineers, Winter Annual Meeting, Chicago, Ill*, pages 126–133, 1980.

[10] M.W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, 2006.

[11] M. von Wattenwyl, M. Clerici, and H. Brauchli. Independent Hybrid Force/Motion Control of Constrained Six-Degrees-of-Freedom Manipulators. *Multibody System Dynamics*, 6(4):327–342, 2001.

[12] D.L. Wedel and G.N. Saridis. An experiment in hybrid position/force control of a six DOFrevolute manipulator. In *1988 IEEE International Conference on Robotics and Automation, 1988. Proceedings.*, pages 1638–1642, 1988.

[13] T. Yoshikawa. Dynamic hybrid position/force control of robot manipulators - Description of hand constraints and calculation of joint driving force. *Robotics and Automation, IEEE Journal of [legacy, pre-1988]*, 3(5):386–392, 1987.

[14] T. Yoshikawa. *Foundations of robotics: analysis and control*. The MIT Press, 1990.