NTNU – Trondheim
Norwegian University of
Science and Technology

Underwater Robotics

control of marine manipulator-vehicle
systems

Simen Andresen

Master of Science in Cybernetics and Robotics
Submission date: June 2014
Supervisor: Kristin Ytterstad Pettersen, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for informasjonsteknologi, matematikk og elektroteknikk
Institutt for teknisk kybernetikk

HOVEDOPPGAVE

Kandidatens navn: **Simen Andresen**

Fag: **Teknisk kybernetikk**

Oppgavens tittel (norsk): Undervannsrobotikk – styring av marine manipulator-AUV/ROV-systemer

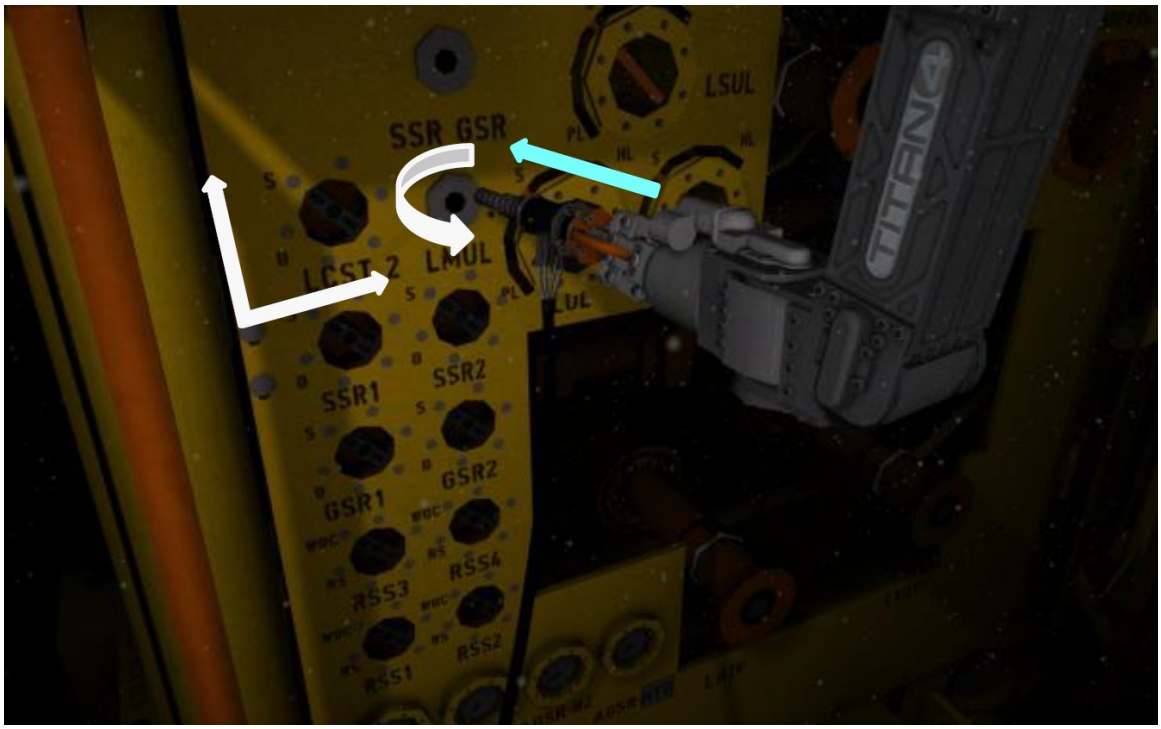
Oppgavens tittel (engelsk): Underwater robotics - control of marine manipulator-vehicle systems

Oppgavens tekst:

Ocean space research using underwater robotics is important for mapping, characterization and monitoring of climate and environment, exploration and exploitation of hydrocarbons and other minerals and resources in demanding areas such as deep water and under ice. The main challenge is to increase the level of autonomy and robustness for automatic mapping, monitoring and intervention, high-level planning/re-planning and reconfiguration of single and multiple vehicles subject to the particular mission, environmental condition, available energy, communication constraints, and any failure conditions.

The Centre of Excellence Autonomous Marine Operations and Systems (AMOS) is dedicated to address these control challenges. The center has high expertise and cutting edge experimental facilities to solve the corresponding control problems. This MSc project will be integrated in the AMOS research, with the corresponding access to cutting edge expertise and experimental facilities.

In particular, this MSc project will address the control challenges of underwater robotics. Development of control and optimisation methods for high-level planning/re-planning and reconfiguration of autonomous underwater vehicles with manipulation capabilities operating in various environmental conditions and in confined areas. This involves modelling of the hydrodynamic loads induced on vehicles by the interaction with the sea bottom, risers or other marine units. Varying hydrodynamic models, both in structure and parameters, and their influence on control system design for vehicle and robotic manipulators will be studied. This is necessary due to different tool configurations and modes of operations. For certain repair operations the underwater vehicle may be connected to the subsea template by some umbilical for energy supply and possible tele-robotics functions.



The goal of this project is to develop improved control methods for automated hot stab operations.

The following subtasks are proposed for this project:

1. Describe the dynamics of the AUV-manipulator system, including the effect of the ocean current.
2. Based on the dynamics model of an AUV-manipulator, develop a control strategy which takes into account the ocean current and uncertainties in the dynamics parameters.
3. Develop a kinematic controller for the AUV-manipulator system for facilitating human operation.
4. Show the effectiveness of the developed control strategies through simulations.

Oppgaven gitt: 06.01.2014

Besvarelsen leveres: 02.06.2014

Besvarelsen levert:

Utført ved Institutt for teknisk kybernetikk

Medveileder: PhD candidate MSc Signe Moe, NTNU

Trondheim, den 06.01.2014

Kristin Y. Pettersen

Faglærer

Preface

This paper constitutes the Master's thesis which is a part of a Master's Degree in engineering cybernetics at the Norwegian University of Science and Technology. It is written during the spring of 2014, and is done in collaboration with the Center of Excellence Autonomous Marine Operations and Systems (AMOS) as a part of their research on underwater autonomy.

Trondheim 2014-06-01

Simen Andresen

Abstract

For using underwater vehicle-manipulator systems (UVMS) in a challenging environment, it is important to have a good mathematical description of the system which accounts for disturbances such as ocean currents. The dynamics equation on matrix form is therefore derived and different properties such as positive definiteness, boundedness and skew symmetry are obtained. Based on the derived equations, a sliding mode controller has been designed in order to track trajectories in the configuration space of the UVMS. The controller is robust when it comes to uncertainties in dynamics parameters and uncertainties in ocean current, yielding global asymptotic stability as long as the uncertainties are bounded.

Furthermore, a kinematic control system has been designed for facilitating human operation of a UVMS, by allowing an operator to only control the end effector motion. The rest of the motion is then resolved through a weighted least-norm pseudo inverse solution of the Jacobian matrix, in order to avoid mechanical joint limits. Moreover, the vehicle's motion is controlled by an event based algorithm to limit the motion of the vehicle. This is done by attaching a 3D meshed polygon to the vehicle frame and check if the end effector is inside or outside this mesh. The mesh then represents the space, relative to the manipulator, were the end effector is fully dexterous. The vehicle will then be commanded to move only when the end effector reaches the outside of the meshed polygon.

A simulator has been implemented, based on the derived equations. The simulations of the UVMS, with the two controllers, yields good tracking results for tracking trajectories both in the workspace of the end effector and in the configuration space of the UVMS.

Sammendrag

For å bruke marine manipulator-AUV/ROV-systemer i et utfordrende miljø er det viktig å ha en god matematisk beskrivelse av systemet som inkluderer ytre påvirkning fra blant annet havstrømmer. Dynamikk-likningene på matrise-form er derfor utledet, og det er vist at de forskjellige leddene har egenskaper som skeiv-symmetri, begrensinger og positiv definitthet. En sliding-mode-regulator har blitt designet, basert på de utledede dynamikk-likningene, for å følge baner i konfigurasjons-rommet til systemet. Regulatoren er robust når det kommer til usikkerhet i dynamikkparametrene og usikkerhet i havstrømmen. Det er også vist at regulatoren gir global asymptotisk stabilitet, så lenge parametrene er avgrenset.

En kinematisk regulator har også blitt laget for å forenkle styring av marine manipulator-AUV/ROV-systemer, ved å la en operator bare styre bevegelsen til manipulatorens gripearm. Resten av bevegelsen til systemet blir da bestemt gjennom en vektet minste-norm pseudo-invers løsning av Jacobian-matrisen. Dette løser problemet med mekaniske begrensinger i manipulator-leddene. Videre blir AUV/ROV-bevegelsen bestemt av en hendelse-basert algoritme som begrenser bevegelsen til AUV/ROV-en så lenge manipulatoren kan bevege seg fritt. Dette blir gjort ved å feste et 3D polygon-mesh til ROV/AUV-rammen og sjekke om gripearmen er innenfor polygon-meshet. ROV/AUVen skal da bare bevege seg når gripearmen når utsiden av polygon-meshet.

En simulator har blitt implementert, basert på de utledede likningene. De to regulatorene viser gode bane-følging-egenskaper, både ved følging av baner direkte i konfigurasjons-rommet til systemet, og av baner for gripearmen til manipulatoren.

Acknowledgment

Firstly, I would like to thank my supervisor, Professor Kristin Y. Pettersen, and co-supervisor, Signe Moe, for giving me the opportunity to work on a topic I find so interesting, and for taking the time to have discussions as well as letting me be free to take the direction I wanted.

Secondly, I would like to thank Torstein A. Myhre for help in the field of robotics.

Thirdly, I would like to thank Nina F. Lillemoen, Hans Erik Frøyen and Andreas Hofman, who I share my office with, for discussions and feedback.

Finally, I would like to thank my family for their love and support.

S. A.

Contents

List of Figures

Notation and Acronyms

Acronyms

AUV	Autonomous Underwater Vehicle
DH	Denavit-Hartenberg
DOF	Degree Of Freedom
ROV	Remotely Operated Vehicle
UVMS	Underwater Vehicle-Manipulator System

Mathematical Notation

n	Number of links of the manipulator
m	Number of degrees of freedom of the vehicle
$\mathcal{F}_0$	Inertial frame
$\mathcal{F}_i$	Frame attached to the UVMS
$\boldsymbol{J} \in \mathbb{R}^{6\times(m+n)}$	Geometrical Jacobian
$\boldsymbol{\omega}_{0b} \in \mathbb{R}^3$	Rotational velocity of vehicle as observed from vehicle frame $\mathcal{F}_b$
$\boldsymbol{R}_{ab} \in \mathbb{R}^{3\times3}$	Rotation matrix representing the rotation of $\mathcal{F}_b$ with respect to $\mathcal{F}_a$
$\boldsymbol{\Theta} = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^\top \in \mathbb{R}^{3\times1}$	Orientation of a frame in Euler angles (roll, pitch and yaw) using the $z-y-x$ convention
$\boldsymbol{J}^\dagger \in \mathbb{R}^{(m+n)\times6}$	Pseudo-inverse of $\boldsymbol{J}$
$\boldsymbol{\nu} \in \mathbb{R}^6$	Body velocity of vehicle
$\boldsymbol{\nu}_r \in \mathbb{R}^6$	Body velocity of vehicle relative to the surrounding water
$\boldsymbol{\nu}_d \in \mathbb{R}^6$	Desired body velocity of vehicle
$\boldsymbol{V}_{0c} = \boldsymbol{V}_{0c}^B \in \mathbb{R}^6$	Twist of $\mathcal{F}_c$ with respect to $\mathcal{F}_0$ as seen from $\mathcal{F}_c$
$\boldsymbol{V}_{0c}^S \in \mathbb{R}^6$	Spatial velocity of $\mathcal{F}_c$

$\boldsymbol{V}_{0e} = \boldsymbol{V}_{0e}^{B} \in \mathbb{R}^6$	Body velocity twist of end effector
$\boldsymbol{\tau} \in \mathbb{R}^{m+n}$	UVMS generalized forces
$\boldsymbol{\tau}_c \in \mathbb{R}^{m+n}$	Commanded generalized forces
$\boldsymbol{\eta} = \begin{bmatrix} (\boldsymbol{p}_{0b})^\top & (\boldsymbol{Q}_{0b})^\top \end{bmatrix}^\top \in \mathbb{R}^6$	Position and rotation (quaternion) of the vehicle, relative to an inertial frame
$\boldsymbol{\eta}_{0e} \in \mathbb{R}^6$	Position and rotation of end effector relative to an inertial frame $\mathcal{F}_0$
$\boldsymbol{q} \in \mathbb{R}^6$	Manipulator joint angles
$\boldsymbol{\xi} = \begin{bmatrix} (\boldsymbol{\eta})^\top & (\boldsymbol{q})^\top \end{bmatrix}^\top \in \mathbb{R}^{m+n}$	Configuration of UVMS
$\boldsymbol{\zeta} = \begin{bmatrix} (\boldsymbol{V}_{0b})^\top & (\dot{\boldsymbol{q}})^\top \end{bmatrix}^\top \in \mathbb{R}^{m+n}$	System velocities of the UVMS
$\boldsymbol{\zeta}_r = \begin{bmatrix} (\boldsymbol{V}_r)^\top & (\dot{\boldsymbol{q}})^\top \end{bmatrix}^\top \in \mathbb{R}^{m+n}$	System velocities relative to the surrounding water
$\boldsymbol{J}_a \in \mathbb{R}^{(m+n)\times(m+n)}$	Analytical Jacobian. Mapping velocities of a frame to the time derivative of the position and orientation coordinates
$\boldsymbol{J}_{a,s} \in \mathbb{R}^{(m+n)\times(m+n)}$	Analytical System Jacobian. Mapping system-velocities to the time derivative of the coordinates $\boldsymbol{\xi}$
$\boldsymbol{g}_{0i} \in \mathbb{R}^{4\times4}$	Homogeneous transformation matrix
$\boldsymbol{Ad}_g \in \mathbb{R}^{6\times6}$	Adjoint transformation matrix associated with the homogeneous transformation $\boldsymbol{g}$
$\boldsymbol{J}_{i0} \in \mathbb{R}^{6\times(m+n)}$	Geometrical Jacobian for body velocities of frame $\mathcal{F}_i$
$\boldsymbol{J}_{e0} \in \mathbb{R}^{6\times(m+n)}$	Geometrical Jacobian for body velocities of end effector
$\|\|\cdot\|\|$	The euclidean norm operator
$diag\{\boldsymbol{p}\}$	A diagonal matrix of size n by n with $\boldsymbol{p} \in \mathbb{R}^n$ along the diagonal
$\widehat{(\,\cdot\,)}$	Skew-symmetric operator

1 | Introduction

1.1 Background

In recent years, technological advancement and an increase in interest for underwater resources have contributed to an increase in unmanned underwater activity, both for research and industry. Examples of this is the extraction of oil and gas from reservoirs under the seabed, marine archeology, and underwater mining. In many aspects of underwater activity, manned operation is considered difficult, unsafe, inefficient and/or tedious. Therefore, underwater robotic systems provide a preferable and in many cases necessary tool.

Remotely Operated Vehicles (ROV) have been used for several decades, and consist of an underwater vehicle, tethered to a manned control station (e.g. a ship) to provide communication, power and to close the human-vehicle control loop.

Autonomous Underwater Vehicles (AUV), on the other hand, should be completely autonomous, and without a tether. An Underwater Vehicle-Manipulator System (UVMS) is a collective term used for both AUV's and ROV's with manipulator capabilities (robot arms). Since ROVs and AUVs share many of their properties that are interesting for modeling and control, UVMS will be the generalization which will be used in this paper.

Today, most ROV-manipulator systems are operated by two people, one operating the ROV, and one operating the manipulator arm. It is reported that it is difficult to find skilled operators, and thus it would be favorable to reduce the number of operators to only one person, and also make the operation simpler. To do this, it is important with a good mathematical description of the system, as well as robust control methods which gives performance under challenging conditions. A

UVMS is a highly dynamical system with complicated kinematics as well as complex dynamics due to high coupling between rigid bodies, hydrodynamics and external influences such as ocean current. In the literature of UVMSs, the ocean current is often left out when describing the total dynamics of the UVMS, and thus, most control laws designed for UVMSs are derived without taking the ocean current into account.

1.2 Related Work

Over the recent decades, much work has been done in the field of underwater robotics. Schjølberg and Fossen [1994] present the dynamics of submerged rigid bodies as well as a method for obtaining the total dynamics, through an iterative Newton-Euler algorithm. It is then proposed to evaluate the iterative algorithm symbolically in order to derive the closed form dynamics equations. Due to the high number of DOFs, it is very difficult to evaluate the iterative algorithm symbolically in order to get meaningful closed form equations that can be used in a control system.

Antonelli [2013] also uses the iterative Newton-Euler algorithm to solve the dynamics of the system. The text presents an outline of the closed form equations of the system, including the ocean current. However, a full mathematical description of the total dynamics equation is not presented in closed form. Antonelli [2013] also presents methods in kinematic control for distributing the commanded motion of the end effector between the vehicle and manipulator, and presents the notion of primary and secondary kinematic tasks that is used in this paper. Interesting results are also presented on adaptive control, and on set point tracking, using Sliding Mode Control (SMC).

From et al. [2013] presents the total kinematics and dynamics equations of a general vehicle-manipulator system. The dynamics equation is derived using Lagrangian mechanics, to obtain the total vehicle-manipulator dynamics equations in matrix form.

Kim et al. [2003] proposes a two-time scale control of a UVMS, where the manipulator and vehicle's different time-scales are utilized. Here the controller is separated

into a slow-dynamics part for the vehicle and a fast-dynamics part for the manipulator. Good tracking results are presented from simulations using a ROV with a 3-link manipulator.

1.3 Problem Formulation

In order to use a UVMS in a challenging environment, it is essential to have a good mathematical description of the system that includes the external influences. It is also essential to have control systems that is robust when it comes to uncertainties in the dynamics parameters. In this paper we want to derive the total dynamics equations for the UVMS, including the influence of the ocean current. We will also present a non-linear control law that is robust when it comes to uncertainties in the system parameters, as well as being robust when faced with an unknown ocean current.

To facilitate human operation of a UVMS, it is necessary to have a high-level control system that make operation of a UVMS intuitive, and easy. We therefore want to obtain a method for commanding the end effector of the manipulator, while the rest of the UVMS's commanded motion is decided by a localized, on-line kinematic control system.

1.4 Outline and Notation

It is important to have a precise notation in order to describe the UVMS mathematically. All vectors are column vectors and are written in bold text, e.g. $\boldsymbol{p}$. All matrices are also written in bold with capital letters, e.g. $\boldsymbol{A}$. Scalars are always represented with a lower case, non-bold notation, e.g. α. Most vectors are written with both subscripts and superscripts to indicate which frames the position or velocity are describing, and which frame it is denoted in. The superscripts are, however, sometimes neglected for notational simplicity, but it should still be clear which frames they are denoted in. This will be discussed later. Mathematical definitions are written with a colon and equality sign, where $a := b$ means that a is by definition equal to b.

When describing the different parts of the UVMS, we will refer to the ROV or AUV, simply as the *vehicle*. The robot arm attached to the vehicle is referred to as the *manipulator*.

We will start by modeling the UVMS in terms of rigid body kinematics and dynamics. In the subsequent chapter, control methods are proposed for the system. A control law for Sliding Mode Control of the system configuration is derived and presented. In the same chapter, a kinematic control system is presented. Furthermore, the results from the simulations of tracking scenarios using the two control laws are given in the next chapter. The last chapter will give a conclusion of the presented work together with suggestions for further work. A CD, containing the simulator software, is attached to the paper. The same software can, however, be downloaded from: https://github.com/simena86/Simulink-Underwater-Robotics-Simulator.

2 | Modeling

In this section, mathematical models of the UVMS are presented. When modeling robotics and marine systems for simulation and control it is customary to look at both the dynamics and the kinematics of the system, and we will therefore look at each of these concepts individually. The kinematics describes the system in terms of geometry and the dynamics describes the relationship between the forces applied to the system and the resulting motion.

The kinematics of the UVMS is highly complex due to the mix of Euclidean and non-Euclidean transformations[1], and a high number of DOFs of the total system. The dynamics is also highly complex, due to the high amount of parameters in the hydrodynamics, strong coupling forces between the rigid bodies, varying inertia for different manipulator configuration and external influences such as sea current.

2.1 Rigid Body Kinematics

Kinematics describes the motion and configuration of a system in terms of geometry, without taking into account how the motion is created. The vehicle and the links of the robot manipulator is considered as rigid bodies, and we therefore present some basic concepts of rigid body kinematics.

2.1.1 Reference Frames

In order to describe mathematically the motion of a rigid body, we will attach coordinate systems to each rigid body. The origin of the coordinate can then be

[1] A transformation is Euclidean if it can be parametrized by generalized coordinates and generalized velocities (From et al. [2013])

used to describe the position of a rigid body relative to another coordinate system, either attached to a rigid body or an inertial reference frame. $\mathcal{F}_0$ is used to denote the earth fixed frame, and is considered to be an inertial frame.[2] Furthermore, $\mathcal{F}_a$ is used to denote some other frame attached to the UVMS. The position of $\mathcal{F}_a$ relative to some $\mathcal{F}_b$ is then denoted $\boldsymbol{p}_{ba}$. To represent the orientation of a frame relative to another, we will use both the rotation matrix and the unit quaternion. The rotation matrix representing the orientation of $\mathcal{F}_a$ relative to $\mathcal{F}_b$ is written $\boldsymbol{R}_{ba}$, can transform a velocity $\boldsymbol{v}^a$ written in $\mathcal{F}_a$ to a velocity written in $\mathcal{F}_b$:

$$\boldsymbol{v}^b = \boldsymbol{R}_{ba}\boldsymbol{v}^a$$

The rotation matrix and the quaternion are used in the implementation of the dynamics and control system to give a singularity free representation. For illustrating the orientation however, we use the Euler angles $\boldsymbol{\Theta} = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}$. The Euler angles represents three consecutive rotations around the z, y, and x axis of the rotated frame. This sequence is known as the roll-pitch-yaw angles and are customary in modeling of ships and aeronautics.

The linear and angular velocity of $\mathcal{F}_a$ relative to $\mathcal{F}_b$ is written $\boldsymbol{v}_{ba}$ and $\boldsymbol{\omega}_{ba}$ respectively. $\boldsymbol{v}_{ba}$ and $\boldsymbol{\omega}_{ba}$ then represents the velocity as observed from $\mathcal{F}_a$, also called the *body velocity*. When written with respect to some other frame $\mathcal{F}_c$, we write the velocities as $\boldsymbol{v}^c_{ba}$ and $\boldsymbol{\omega}^c_{ba}$.

The *homogeneous transformation* matrix is used to represent the position and rotation of a frame relative to another[3]. The matrix is defined as

$$\boldsymbol{g}_{ab} := \begin{bmatrix}\boldsymbol{R}_{ab} & \boldsymbol{p}_{ab} \\ \boldsymbol{0}_{1\times 3} & 1\end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{2.1}$$

Where $\boldsymbol{R}_{ab}$ is the rotation matrix from $\mathcal{F}_a$ to $\mathcal{F}_b$, and $\boldsymbol{p}_{ab}$ is the vector representing the linear displacement of the origin of $\mathcal{F}_b$ with respect to $\mathcal{F}_a$. The homogeneous transformation matrix belongs to the special Euclidean group $SE(3)$ and can be seen as a map from one coordinate system into another (Murray et al. [1994]).

[2]Although earth fixed frames are non-inertial, when moving at low speed this is a good approximation, see e.g. Fossen [2011] for a discussion.

[3]Throughout this text, the term pose are sometimes used to describe both the position and the rotation of a frame

2.1.2 Quaternion Orientation Representation

The orientation of a rigid body can be described by a unit quaternions $\boldsymbol{Q}$ which belongs to the set $\mathbb{H}$ defined by (Chou [1992])

$$\mathbb{H} = \left\{ \boldsymbol{Q} | \boldsymbol{Q}^\top \boldsymbol{Q} = 1, \boldsymbol{Q} = \left[\eta, \boldsymbol{\epsilon}^\top \right]^\top \right\} \tag{2.2}$$

The scalar part η and the vector part $\boldsymbol{\epsilon}$ of a quaternion represents a rotation by an angle β around a unit axis $\boldsymbol{\lambda}$, described by the following:

$$\eta = cos(\frac{\beta}{2}) \tag{2.3}$$

$$\boldsymbol{\epsilon} = \boldsymbol{\lambda} sin(\frac{\beta}{2}) \tag{2.4}$$

It is a two-to-one correspondence between $\mathbb{H}$ and $SO(3)$ because $\boldsymbol{Q} = -\boldsymbol{Q}$ (Fjellstad and Fossen [1994c]) which is usually solved by choosing the quaternion $\boldsymbol{Q}$ such that $\eta \geq 0$. We also define the complex conjugate of a quaternion $\boldsymbol{Q}$ as:

$$\bar{\boldsymbol{Q}} := \begin{bmatrix} \eta \\ -\boldsymbol{\epsilon} \end{bmatrix} \tag{2.5}$$

The error quaternion $\widetilde{\boldsymbol{Q}}$ representing the error between quaternion $\boldsymbol{Q}_d$ and $\boldsymbol{Q}$, which will be used in the control design, can then be described by taking the quaternion product between the $\bar{\boldsymbol{Q}}_d$ and $\boldsymbol{Q}$

$$\widetilde{\boldsymbol{Q}} := \begin{bmatrix} \widetilde{\eta} \\ \widetilde{\boldsymbol{\epsilon}} \end{bmatrix} \tag{2.6}$$

$$= \begin{bmatrix} \eta\eta_d + \boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}_d \\ \eta\boldsymbol{\epsilon}_d - \eta_d\boldsymbol{\epsilon} + \widehat{\boldsymbol{\epsilon}}_d\boldsymbol{\epsilon} \end{bmatrix} \tag{2.7}$$

Furthermore, we define the transformation between body angular velocity and the time derivative of $\boldsymbol{Q}$ as

$$\dot{\boldsymbol{Q}} = \boldsymbol{T}_Q(\boldsymbol{Q})\boldsymbol{\omega} \tag{2.8}$$

$$\boldsymbol{\omega} = \boldsymbol{T}_Q(\boldsymbol{Q})^\top \dot{\boldsymbol{Q}} \tag{2.9}$$

where (2.10)

$$\boldsymbol{T}_Q(\boldsymbol{Q}) = \frac{1}{2} \begin{bmatrix} -\boldsymbol{\epsilon}^\top \\ \eta\boldsymbol{I}_{3\times 3} + \widehat{\boldsymbol{\epsilon}} \end{bmatrix} \tag{2.11}$$

The error dynamics can be described by the following (Fjellstad and Fossen [1994c])

$$\dot{\widetilde{\boldsymbol{Q}}} = \boldsymbol{T}_Q(\widetilde{\boldsymbol{Q}})(\boldsymbol{Q}_{0a})\widetilde{\boldsymbol{\omega}} \tag{2.12}$$

where $\widetilde{\boldsymbol{\omega}} = \boldsymbol{\omega}_d - \boldsymbol{\omega}$ is the difference between some desired and measured angular velocity.

2.1.3 Twists

The notion of twists gives a compact way of representing the velocity of a frame with respect to another. The twist $\boldsymbol{V}$ representing the velocity between two frames can be written as

$$\boldsymbol{V}_{ab} := \begin{bmatrix} \boldsymbol{v}_{ab}^\top & \boldsymbol{\omega}_{ab}^\top \end{bmatrix}^\top \in \mathbb{R}^6 \tag{2.13}$$

where the two stacked vectors $\boldsymbol{v}_{ab}$ and $\boldsymbol{\omega}_{ab}$ represent the linear and angular velocity of frame $\mathcal{F}_b$ relative to $\mathcal{F}_a$. A twist can also be written with respect to another frame. In this text a twist $\boldsymbol{V}_{ab}$ will represent the velocity as seen from $\mathcal{F}_b$, which is called the *body twist*. To avoid ambiguity, the body twist is also sometimes expressed explicitly with a superscripted capital B: $\boldsymbol{V}_{ab}^B$. If the twist is represented in some other frame, it will be written with superscript. For example, $\boldsymbol{V}_{ab}^0$ is the velocity between $\mathcal{F}_a$ and $\mathcal{F}_b$ as seen from $\mathcal{F}_0$. A twist can also be represented as a *spatial twist*. Spatial twists are written with superscript S and are defined somewhat different than body twists: the linear part of the spatial velocity of a rigid body relative to a reference frame is the velocity of a point attached to a possible imaginary extension of the body, moving through the reference frame (From et al. [2013]). The angular velocity part of the spatial velocity is the angular velocity as seen from the reference frame. Although somewhat non-intuitive, the spatial representation facilitates modeling of a multi-body system (see Featherstone [2010] or Murray et al. [1994]). The spatial twist is also used in the derivation of the velocity kinematics.

The *adjoint transformation* associated with the homogeneous transformation

matrix $\boldsymbol{g}_{ab}$, is represented by the matrix $\boldsymbol{Ad}_{g_{ab}}$, and is defined by the following:

$$\boldsymbol{Ad}_{g_{ab}} := \begin{bmatrix} \boldsymbol{R}_{ab} & \widehat{\boldsymbol{p}}_{ab}\boldsymbol{R}_{ab} \\ \boldsymbol{0} & \boldsymbol{R}_{ab} \end{bmatrix} \in \mathbb{R}^{6\times 6} \tag{2.14}$$

$$\boldsymbol{Ad}^{-1}_{g_{ab}} = \begin{bmatrix} \boldsymbol{R}^{\top}_{ab} & -\boldsymbol{R}^{\top}_{ab}\widehat{\boldsymbol{p}}_{ab} \\ 0 & \boldsymbol{R}^{\top}_{ab} \end{bmatrix} \in \mathbb{R}^{6\times 6} \tag{2.15}$$

where the hat operator $\widehat{(\cdot)}$ maps a vector to its skew-symmetric matrix representation, and when operating on a vector $\boldsymbol{\omega} \in \mathbb{R}^3$ is defined as

$$\widehat{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{2.16}$$

The adjoint transformation matrix can map a body twist to a spatial twist

$$\boldsymbol{V}^S_{ab} = \boldsymbol{Ad}_{g_{ab}}\boldsymbol{V}_{ab} \tag{2.17}$$

It also has the following properties (Murray et al. [1994]):

$$\boldsymbol{Ad}_{g_{ac}} = \boldsymbol{Ad}_{g_{ab}}\boldsymbol{Ad}_{g_{bc}} \tag{2.18}$$

$$\boldsymbol{Ad}^{-1}_{g_{ac}} = \boldsymbol{Ad}_{g_{ca}} \tag{2.19}$$

Remark 1. *If we let $\mathcal{F}_c$ and $\mathcal{F}_d$ be attached to the same rigid body, and let $\boldsymbol{V}_{0c}$ and $\boldsymbol{V}_{0d}$ be the body velocities of each frame, we can use the adjoint transformation matrix to map between the two:*

$$\boldsymbol{V}_{0c} = \boldsymbol{Ad}_{g_{cd}}\boldsymbol{V}_{0d} \tag{2.20}$$

Proof. We have the following

$$\boldsymbol{V}^S_{0c} = \boldsymbol{Ad}_{g_{0c}}\boldsymbol{V}_{0c} \tag{2.21}$$

$$\boldsymbol{V}^S_{0d} = \boldsymbol{Ad}_{g_{0d}}\boldsymbol{V}_{0d} \tag{2.22}$$

From the definition of a spatial velocity twist, we see that the spatial velocity of a rigid body is not depending on a specific frame of the body, and we have that

$$\boldsymbol{V}^S_{0c} = \boldsymbol{V}^S_{0d} \tag{2.23}$$

We can then substitute (2.21) and (2.22) and use the properties in (2.19) and (2.18) to get:

$$\boldsymbol{Ad}_{g_{0c}}\boldsymbol{V}_{0c} = \boldsymbol{Ad}_{g_{0d}}\boldsymbol{V}_{0d} \tag{2.24}$$

$$\boldsymbol{V}_{0c} = \boldsymbol{Ad}^{-1}_{g_{0c}}\boldsymbol{Ad}_{g_{0d}}\boldsymbol{V}_{0d} \tag{2.25}$$

$$\boldsymbol{V}_{0c} = \boldsymbol{Ad}_{g_{c0}}\boldsymbol{Ad}_{g_{0d}}\boldsymbol{V}_{0d} \tag{2.26}$$

$$\boldsymbol{V}_{0c} = \boldsymbol{Ad}_{g_{cd}}\boldsymbol{V}_{0d} \tag{2.27}$$

□

2.1.4 Velocity Transformations

A problem when describing the velocity of a rigid body is that it cannot be described with *general velocities* corresponding to some general coordinates. *Generalized coordinates* is, in this context, explained as a set of coordinates uniquely describing the configuration of a system relative to some reference configuration. Generalized coordinates are used to describe the geometrics of the vehicle position and orientation, as well as the joint angles of the manipulator. *General velocities* are the time derivative of the general velocities. *Quasi-velocities* on the other hand are velocities of a system that in general are not equal to the time derivative of the generalized coordinates. An example of quasi-velocities is the body angular velocity $\boldsymbol{\omega}$. There does not exist any general coordinates $\boldsymbol{x}$ for the orientation of a rigid body that gives us $\boldsymbol{\omega} = \dot{\boldsymbol{x}}$.

For any quasi-velocity $\boldsymbol{\gamma}$ the time derivative of the generalized coordinates $\boldsymbol{x}$ then have to be mapped to the quasi-velocities $\boldsymbol{\gamma}$ through $\boldsymbol{\gamma} = \boldsymbol{S}(\boldsymbol{x})\dot{\boldsymbol{x}}$. This mapping is described using the analytical Jacobian(From et al. [2013]):

$$\boldsymbol{J}_a = \boldsymbol{S}(\boldsymbol{x})^{-1} \tag{2.28}$$

When using generalized coordinates with for instance Euler angles as the representation of orientation of a rigid body, this transformation will be singular for certain configurations. This is one of the motivations to use quaternions, as this gives no singularities at the cost of using one extra parameter to describe the orientation.

For describing the orientation and position of a frame, we then use a 7-parameter vector $\boldsymbol{\eta}$:

$$\boldsymbol{\eta} := \begin{bmatrix} \boldsymbol{p}^\top & \boldsymbol{Q}^\top \end{bmatrix}^\top$$

where $\boldsymbol{p}$ is the position and $\boldsymbol{Q}$ is the quaternion orientation representation, relative to some frame. The corresponding quasi-velocity is then a twist $\boldsymbol{V}$. For a frame $\mathcal{F}_b$ moving relative to $\mathcal{F}_0$ we can then describe the transformation between the twist $\boldsymbol{V}_{0b}$, and the derivative of the coordinates $\boldsymbol{\eta}$:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_a \boldsymbol{V}_{0b}$$

where (2.29)

$$\boldsymbol{J}_a = \begin{bmatrix} \boldsymbol{R}_{0b} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}_Q(\boldsymbol{Q}_{0b}) \end{bmatrix} \tag{2.30}$$

2.2 UVMS Kinematics

To derive the kinematics of a UVMS, one assumes that all the links in the system are perfectly rigid, and the motion of each rigid body is composed of either a translation, a rotation or both.

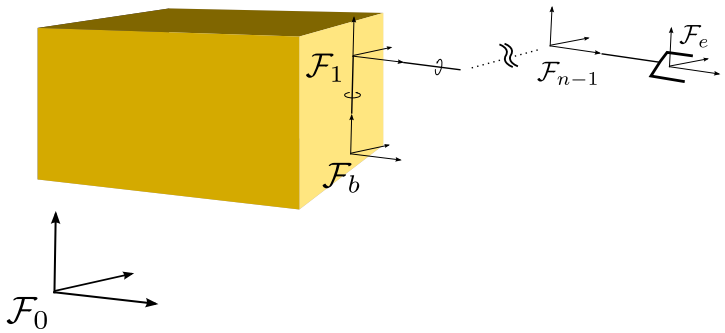


Figure 2.1: The assignment of the frames for the UVMS system

The vehicle is regarded as a rigid body with the normal 6 DOFs, and the manipulator is a 6-link kinematic chain with only 1-DOF revolute joints. To represent the kinematic structure of the UVMS, a number of frames are assigned as illustrated in Fig. 2.1. A reference frame $\mathcal{F}_0$ is attached to the earth and is considered inertial.

The frame $\mathcal{F}_b$ is attached to the vehicle, at the location of the manipulator base. The frames of the manipulator are attached according to the Denavit-Hartenburg (DH) convention (Spong and Hutchinson [2005]). The *end effector* is the tool at the last link of the end effector, and we will therefore refer to $\mathcal{F}_6$ as the end effector frame, and denote it $\mathcal{F}_e$.

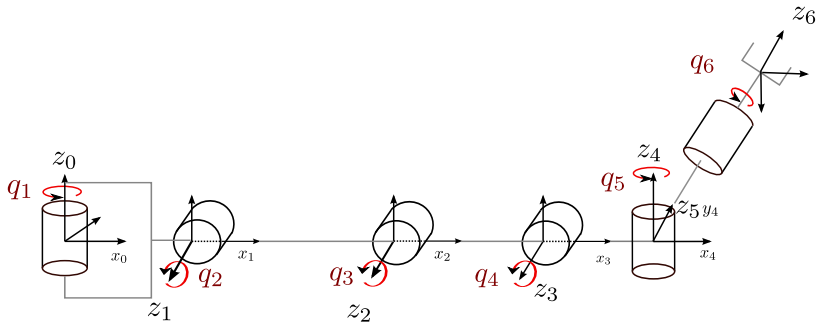


Figure 2.2: The frames are assigned to the kinematic structure of the manipulator according to the DH convention

We use the quaternion $\boldsymbol{Q}_{0b}$ to represent the orientation of the vehicle relative to $\mathcal{F}_0$. The vehicle position is represented using the coordinates of the origin of $\mathcal{F}_b$ relative to $\mathcal{F}_0$. We write the vehicle configuration as:

$$\boldsymbol{\eta} := \begin{bmatrix} \boldsymbol{p}_{0b} \\ \boldsymbol{Q}_{0b} \end{bmatrix} = \begin{bmatrix} x_{0b} \\ y_{0b} \\ z_{0b} \\ \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} \in \mathbb{R}^7$$

The configuration of the manipulator is represented using the vector of joint angles

$$\boldsymbol{q} := \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix}^\top \in \mathbb{R}^6$$

The total configuration can then be written in vector form

$$\boldsymbol{\xi} := \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{q} \end{bmatrix} \in \mathbb{R}^{13}$$

Furthermore, he velocities of the UVMS are defined by:

$$\boldsymbol{\zeta} := \begin{bmatrix} \boldsymbol{\nu} \\ \dot{\boldsymbol{q}} \end{bmatrix} \in \mathbb{R}^{12}$$

where we use $\boldsymbol{\nu} := \boldsymbol{V}_{0b}$ to comply with the notation used in modeling of marine crafts. We will refer to $\boldsymbol{\zeta}$ as the *system velocity*. The system velocity can be mapped to the time derivative of the generalized coordinates through the analytical Jacobian for the total system:

$$\dot{\boldsymbol{\xi}} = \boldsymbol{J}_{a,s}\boldsymbol{\zeta} \tag{2.31}$$

$$\boldsymbol{J}_{a,s} = \begin{bmatrix} \boldsymbol{R}_{0b} & 0 & 0 \\ 0 & \boldsymbol{T}_Q(\boldsymbol{Q}_{0b}) & 0 \\ 0 & 0 & \boldsymbol{I}_{6\times 6} \end{bmatrix} \in \mathbb{R}^{(6+n)\times(6+n)} \tag{2.32}$$

The configuration states are now defined for the total system, together with the mapping between the velocity states and the rate of change of the generalized coordinates through the analytical Jacobian $\boldsymbol{J}_{a,s}$. Furthermore, it is important to define the velocity of frames with respect to an inertial frame, where the velocity of the end effector frame $\mathcal{F}_e$ is the most important. This is done through the geometric Jacobian $\boldsymbol{J}_{i0}$, which maps $\boldsymbol{\zeta} \to \boldsymbol{V}_{0i}$ for any frame $\mathcal{F}_i$ attached to the UVMS.

First, each manipulator link's velocity is represented as a body velocity twist by scaling a unit velocity twist $\boldsymbol{X}_i^i$ by the associated velocity $\dot{q}_i$. Using the notation from From et al. [2013] one can write the body joint twist of a revolute joint as

$$\boldsymbol{X}_i^i = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\top \tag{2.33}$$

Thus, we can write the body velocity of each link as $\dot{q}_i \boldsymbol{X}_i^i$, which represents the velocity of a joint with respect to the frame attached to it. It should be noted that the twist only has one nonzero component in the angular motion around the z-axis due to following the DH convention for revolute joints.

Next, we write the velocity of each joint relative to the frame attached to it in spatial coordinates as $\dot{q}_i \boldsymbol{X}_i$. Recalling that the adjoint transformation matrix can transform body twists to spatial twists, we get the following.

$$\boldsymbol{X}_i = \boldsymbol{Ad}_{g_{bi}} \boldsymbol{X}_i^i \tag{2.34}$$

Using (2.14) and (2.33) yields:

$$\boldsymbol{X}_i = \begin{bmatrix} \boldsymbol{R}_{bi} & \widehat{\boldsymbol{p}}_{bi}\boldsymbol{R}_{bi} \\ \boldsymbol{0} & \boldsymbol{R}_{bi} \end{bmatrix} \boldsymbol{X}_i^i \tag{2.35}$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_2r_{31}-p_3r_{21} & p_2r_{32}-p_3r_{22} & p_2r_{33}-p_3r_{23} \\ r_{21} & r_{22} & r_{23} & -p_1r_{31}+p_3r_{11} & -p_1r_{32}+p_3r_{12} & -p_1r_{33}+p_3r_{13} \\ r_{31} & r_{32} & r_{33} & p_1r_{21}+p_2r_{11} & p_1r_{22}+p_2r_{12} & p_1r_{23}+p_2r_{13} \\ 0 & 0 & 0 & r_{11} & r_{12} & r_{13} \\ 0 & 0 & 0 & r_{21} & r_{22} & r_{23} \\ 0 & 0 & 0 & r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.36}$$

$$= \begin{bmatrix} p_2r_{33}-p_3r_{23} \\ -p_1r_{33}+p_3r_{13} \\ p_1r_{23}+p_2r_{13} \\ r_{13} \\ r_{23} \\ r_{33} \end{bmatrix} \tag{2.37}$$

Since the velocity $\dot{q}_i \boldsymbol{X}_i$ is in spatial coordinates, we can get the spatial velocity of each frame $\mathcal{F}_i$, relative to $\mathcal{F}_b$, by summing up all the velocities further down in the kinematic chain:

$$\boldsymbol{V}_{0i}^S = \sum_{k=1}^{k \leq i} \dot{q}_i \boldsymbol{X}_i \tag{2.38}$$

This can be written as a mapping from the manipulator joint velocities $\dot{\boldsymbol{q}}$ to the spatial velocities $\boldsymbol{V}_{0i}^S$ using the geometric Jacobian:

$$\boldsymbol{V}_{bi}^S = \boldsymbol{J}_i \dot{q} \tag{2.39}$$

where (2.40)

$$\boldsymbol{J}_i = \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{X}_2 & \cdots & \boldsymbol{X}_i & \boldsymbol{0}_{6\times(n-i)} \end{bmatrix} \tag{2.41}$$

Moreover, we want to find a map from the system velocities to the velocities of each frame $\mathcal{F}_i$ with respect to the inertial frame $\mathcal{F}_0$ in spatial coordinates. First, we write the body twist of each frame relative to the inertial frame as a sum (From et al. [2013]):

$$\boldsymbol{V}_{0i} = \boldsymbol{Ad}_{g_{bi}}^{-1} \boldsymbol{V}_{0b} + \boldsymbol{V}_{bi} \tag{2.42}$$

We then use that

$$\boldsymbol{V}_{bi}^{S} = \boldsymbol{Ad}_{g_{bi}} \boldsymbol{V}_{bi} \tag{2.43}$$

$$\Updownarrow$$

$$\boldsymbol{V}_{bi} = \boldsymbol{Ad}_{g_{bi}}^{-1} \boldsymbol{V}_{bi}^{S} \tag{2.44}$$

Inserting (2.39) and (2.44) into (2.42) yields:

$$\boldsymbol{V}_{0i} = \boldsymbol{Ad}_{g_{bi}}^{-1} \boldsymbol{V}_{0b} + \boldsymbol{Ad}_{g_{bi}}^{-1} \boldsymbol{J}_i \dot{\boldsymbol{q}} \tag{2.45}$$

Finally, we can express this using the Jacobian matrix $\boldsymbol{J}_{i0}$

$$\boldsymbol{V}_{0i} = \boldsymbol{J}_{i0} \boldsymbol{\zeta} \tag{2.46}$$

where (2.47)

$$\boldsymbol{J}_{i0} = \begin{bmatrix} \boldsymbol{Ad}_{g_{bi}}^{-1} & \boldsymbol{Ad}_{g_{bi}}^{-1} \boldsymbol{J}_i \end{bmatrix} \tag{2.48}$$

From (2.46) we get that the end effector velocity can be described using the following

$$\boldsymbol{V}_{0e} = \boldsymbol{J}_{e0} \boldsymbol{\zeta} \tag{2.49}$$

$$= \begin{bmatrix} \boldsymbol{Ad}_{g_{be}}^{-1} & \boldsymbol{Ad}_{g_{be}}^{-1} \boldsymbol{J}_n \end{bmatrix} \boldsymbol{\zeta} \tag{2.50}$$

Where $\boldsymbol{J}_n$ is defined in (2.41) with $i = n$.

2.3 Kinematics of the Ocean Current

Some of the forces acting on the UVMS is dependent on the system's velocity relative to the surrounding water rather than the velocity relative to an inertial frame. If the surrounding water is not moving, the system's velocity relative to the water and relative to some inertial frame is the same. However, with the presence of an ocean current the two will in general be different.

As an example, the velocity of the Gulf Stream can reach up to 2.5m/s (Stommel [1958]) , and it is therefore important to have a good description of the kinematics of the ocean current, when deriving the dynamics of the system. In most of the literature of underwater vehicle-manipulator systems, the current is left out when describing the full dynamics equations and when designing control laws. In this paper, however, the current is included to give a more accurate description.

We consider an irotational current described by the linear velocity $\boldsymbol{v}_c^0$, which is constant in the inertial frame $\mathcal{F}_0$. Furthermore, we define the velocity twist of the current

$$\boldsymbol{V}_c^0 := \begin{bmatrix} (\boldsymbol{v}_c^0)^\top & (\boldsymbol{0}_{1\times 3})^\top \end{bmatrix}^\top \tag{2.51}$$

$$\dot{\boldsymbol{V}}_c^0 = 0 \tag{2.52}$$

In order to use the velocity of the ocean current in the dynamics equation, the velocity must be described in the frame of the vehicle which is obtained by rotating $\boldsymbol{v}_c^0$:

$$\boldsymbol{V}_c^b = \begin{bmatrix} \boldsymbol{R}_{0b}^\top \boldsymbol{v}_c^0 \\ \boldsymbol{0}_{3\times 1} \end{bmatrix} \tag{2.53}$$

$$\dot{\boldsymbol{V}}_c^B = \begin{bmatrix} \dot{\boldsymbol{R}}_{0b}^\top \boldsymbol{v}_c^0 + \boldsymbol{R}_{0b}^\top \dot{\boldsymbol{v}}_c^0 \\ \boldsymbol{0} \end{bmatrix} \tag{2.54}$$

Using that $\boldsymbol{R}_{0b}^\top = \boldsymbol{R}_{b0}$, $\boldsymbol{\omega}_{0b} = -\boldsymbol{\omega}_{b0}$, $\dot{\boldsymbol{v}}_c^0 = 0$ and $\dot{\boldsymbol{R}}_{0b} = \boldsymbol{R}_{0b}\widehat{\boldsymbol{\omega}}_{0b}$, (2.54) yields

$$\dot{\boldsymbol{V}}_c^b = \begin{bmatrix} -\boldsymbol{R}_{0b}^\top \widehat{\boldsymbol{\omega}}_{0b} \boldsymbol{v}_c^0 \\ \boldsymbol{0} \end{bmatrix} \tag{2.55}$$

The obtained relative velocity and acceleration of the vehicle can now be used to describe the total velocity relative to the current.

$$\begin{aligned} \boldsymbol{\zeta}_r :&= \begin{bmatrix} \boldsymbol{\nu} - \boldsymbol{V}_c^b \\ \dot{\boldsymbol{q}} \end{bmatrix} \\ &= \boldsymbol{\zeta} - \boldsymbol{H}_m(\boldsymbol{\xi}) \boldsymbol{v}_c^0 \end{aligned} \tag{2.56}$$

where

$$\boldsymbol{H}_m(\boldsymbol{\xi}) = \begin{bmatrix} \boldsymbol{R}_{0b}^\top \\ \boldsymbol{0}_{9\times 3} \end{bmatrix} \in \mathbb{R}^{12\times 3} \tag{2.57}$$

While the relative acceleration yields

$$\dot{\boldsymbol{\zeta}}_r = \begin{bmatrix} \dot{\boldsymbol{\nu}} - \dot{\boldsymbol{V}}_c^b \\ \ddot{\boldsymbol{q}} \end{bmatrix} \tag{2.58}$$
$$= \dot{\boldsymbol{\zeta}} - \boldsymbol{H}_n(\boldsymbol{\xi}, \boldsymbol{\zeta}) \boldsymbol{v}_c^0$$

where

$$\boldsymbol{H}_n(\boldsymbol{\xi}, \boldsymbol{\zeta}) = \begin{bmatrix} -\boldsymbol{R}_{0b}^\top \widehat{\boldsymbol{\omega}}_{0b} \\ \boldsymbol{0}_{9\times 3} \end{bmatrix} \in \mathbb{R}^{12\times 3} \tag{2.59}$$

The total relative velocity $\boldsymbol{\zeta}_r$ only needs the relative velocity of the vehicle, while the joint links remain as in $\boldsymbol{\zeta}$. The relative velocity of the vehicle propagates to the links down the kinematic chain. Since the velocity of a link relative to the surrounding water is a sum of the previous link velocities and the relative velocity of the vehicle, it is clear that the total relative velocity and acceleration can be described with (2.56) and (2.58).

2.4 Dynamics

The dynamics of the UVMS describes the relationship between the forces and the corresponding motion of the system. As customary in modeling of ship dynamics and robotics, the equations of motion are derived from the knowledge of the total energy of the system using the Lagrangian L

$$L = T - V$$

Where T and V is the kinetic and potential energy, respectively. Due to the high number of states of the system, the equations of motion are presented in a matrix form, adopted from the robotics literature, which also has been customary in modern literature on marine craft modeling and control. First, the equations of motion of the sole vehicle is presented, before the total system including the manipulator is described.

2.4.1 Wrenches

A generalized force acting on a rigid body consists of linear and angular components, also referred to as forces and moments. Using a *wrench* we can represent this quantity as a stacked column vector. We define a wrench $\boldsymbol{F}$ as:

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{n} \end{bmatrix} \in \mathbb{R}^6 \tag{2.60}$$

where (2.61)

$$\boldsymbol{f} \in \mathbb{R}^3 \text{ linear component} \tag{2.62}$$

$$\boldsymbol{n} \in \mathbb{R}^3 \text{ angular component} \tag{2.63}$$

A wrench acting on the origin of $\mathcal{F}_i$ is denoted $\boldsymbol{F}_i$. To represent the wrench in a different frame one can use the adjoint transformation matrix(Murray et al. [1994]):

$$\boldsymbol{F}_a = (\boldsymbol{Ad}_{g_{ba}})^\top \boldsymbol{F}_b \tag{2.64}$$

Where $\boldsymbol{F}_a$ and $\boldsymbol{F}_b$ are wrenches written in $\mathcal{F}_a$ and $\mathcal{F}_b$ respectively

2.4.2 Vehicle Dynamics

A model of a marine craft is proposed in Fossen [2011]

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{V}_{0b}^B + \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{g}_0 + \tag{2.65}$$

$$\boldsymbol{M}_A\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r \tag{2.66}$$

$$= \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \tag{2.67}$$

Where $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{V}_c^b$ is the velocity of the vehicle relative to the water surrounding it. Using a parametrization of the Coriolis forces that are not dependent on linear velocity, it can be shown that (Hegrenæs [2010])

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_{RB}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r \tag{2.68}$$

The equation of motion can therefore be written as

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{g}_0 = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \tag{2.69}$$

$$\boldsymbol{M} := \boldsymbol{M}_{RB} + \boldsymbol{M}_A \tag{2.70}$$

$$\boldsymbol{C}(\boldsymbol{\nu}_r) := \boldsymbol{C}_{RB}(\boldsymbol{\nu}_r) + \boldsymbol{C}_A(\boldsymbol{\nu}_r) \tag{2.71}$$

Due to the properties of the Inertia and Coriolis matrix, the total dynamics can be written in the compact form of equation (2.69), where the system velocity is uniquely described by the relative velocity $\boldsymbol{\nu}_r$.

2.4.3 Vehicle-Manipulator Dynamics

When modeling the dynamics of the UVMS, we will assume that it is totally submerged, and therefore the wind will be removed from (2.69). Also, the force from waves is neglected, which is reasonable since the operation will mostly take place in sufficiently deep waters. In the literature of underwater robotics, such as Schjølberg and Fossen [1994] and Antonelli [2013], among others, the total dynamics is derived using the iterative Newton-Euler algorithm. This algorithm first derives the orientation, velocities and accelerations of each frame by outward iterating over the kinematic chain, starting from the vehicle. It then uses the obtained orientations, velocities and accelerations to derive the forces on each link. This is done by iterating backwards over the kinematic chain and summing the forces, starting from the end effector. This approach gives an easy way of simulating the dynamics. However, there is no straight forward way of obtaining the matrix formulation, which is important for the design of a controller. In this paper the total forces are derived by projecting the forces and inertial forces acting on each body in the kinematic chain to the vector of generalized forces $\boldsymbol{\tau}$, using the Jacobian. This is inspired from the work on dynamics in From et al. [2013], but is extended to also include the hydrodynamics.

Projecting the forces on each body is very advantageous since one can use well-known properties of single, rigid bodies moving underwater, which is described in e.g. Fossen [2011], without taking into account the coupling between the bodies. The coupling of the bodies is then accounted for when projecting the forces on each

body to the space of generalized forces $\boldsymbol{\tau}$. Here, the generalized forces are the forces acting in the direction of the system velocity $\boldsymbol{\zeta}$, and is written:

$$\boldsymbol{\tau} := \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 & \tau_5 & \tau_6 & \tau_7 & \tau_8 & \tau_9 & \tau_{10} & \tau_{11} & \tau_{12} \end{bmatrix}^\top$$

We start by describing the wrench which acts on $\mathcal{F}_i$ of the ith body

$$\boldsymbol{F}_i = \begin{bmatrix} f_x^i & f_y^i & f_z^i & n_x^i & n_y^i & n_z^i \end{bmatrix}^\top \tag{2.72}$$

$\boldsymbol{F}_i$ is then a vector of forces and moments acting on $\mathcal{F}_i$ denoted in the same frame. From Spong and Hutchinson [2005] we get that the forces acting on a frame can be mapped to the generalized forces $\boldsymbol{\tau}$ using the Jacobian:

$$\boldsymbol{\tau}_i = (\boldsymbol{J}_{i0})^\top \boldsymbol{F}_i \tag{2.73}$$

Using this relationship, one can get the total generalized force from $\boldsymbol{F}_i, i \in [b, n]$ by summing up all the projected wrenches.

$$\boldsymbol{\tau} = \sum_{i=b}^{n} (\boldsymbol{J}_{i0})^\top \boldsymbol{F}_i \tag{2.74}$$

Furthermore, the hydrodynamical and potential forces are often described in CG (center of gravity) or CB (center of buoyancy) and the hydrodynamical forces are normally derived using the velocity of CG. We will therefore calculate the forces in either CG or CB and map these forces to CO (origin of $\mathcal{F}_i$), and use the velocity of CG in the hydrodynamics. From (2.20) and (2.64) we get that we can use an adjoint transformation matrix to do this mapping. We define the two matrices $\boldsymbol{Ad}_{g_{ici}}$ and $\boldsymbol{Ad}_{g_{ibi}}$ as

$$\boldsymbol{Ad}_{g_{ici}} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \widehat{\boldsymbol{r}_{i,ci}^i} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \ , \ \boldsymbol{Ad}_{g_{ici}}^{-1} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & -\widehat{\boldsymbol{r}_{i,ci}^i} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{2.75}$$

$$\boldsymbol{Ad}_{g_{ibi}} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \widehat{\boldsymbol{r}_{i,bi}^i} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \ , \ \boldsymbol{Ad}_{g_{ibi}}^{-1} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & -\widehat{\boldsymbol{r}_{i,bi}^i} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{2.76}$$

Where $\boldsymbol{r}_{i,ci}^i$ and $\boldsymbol{r}_{i,bi}^i$ are the vectors from the origin of $\mathcal{F}_i$ to CG and CB of link i according to Fig. 2.3. Let $\boldsymbol{V}_{0ci}$ be the body velocity of the center of CG of link

i, and let $\boldsymbol{F}_{ci}$ and $\boldsymbol{F}_{bi}$ be the wrenches acting on CG and CB respectively. From (2.20) and (2.64) together with the geometrical jacobian we then get the following transformations

$$\begin{aligned} \boldsymbol{V}_{0ci} &= \boldsymbol{Ad}_{g_{ici}}^{-1} \boldsymbol{V}_{0i} && (2.77) \\ &= \boldsymbol{Ad}_{g_{ici}}^{-1} \boldsymbol{J}_{i0} \boldsymbol{\zeta} \\ \boldsymbol{F}_i &= (\boldsymbol{Ad}_{g_{ici}})^{-\top} \boldsymbol{F}_{ici} && (2.78) \\ \boldsymbol{F}_i &= (\boldsymbol{Ad}_{g_{ibi}})^{-\top} \boldsymbol{F}_{ibi} && (2.79) \end{aligned}$$

The transformations above, together with the mapping of the wrenches into the

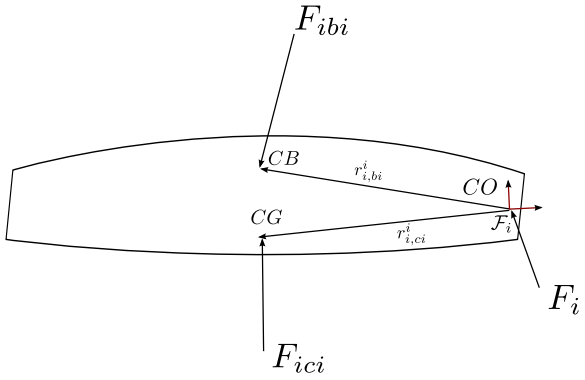


Figure 2.3: Forces on a generic body of the UVMS.

generalized forces in (2.74) will now be used to derive the total dynamics for the system.

2.4.3.1 Inertial Forces

The inertial forces acting on CO of body i of the UVMS comes from acceleration and rotating the rigid body with respect to $\mathcal{F}_0$. We let m denote the mass of a link, and let $\boldsymbol{I}_g$ denote the moment of Inertia matrix(Fossen [2011]):

$$\boldsymbol{I}_g := \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix}$$

We can now write the inertial forces of each of the rigid bodies of the UVMS in a matrix formulation:

$$\boldsymbol{F}_i = \boldsymbol{M}_{i,RB}\dot{\boldsymbol{V}}_{0i}^i + \boldsymbol{C}_{i,RB}(\boldsymbol{V}_{0i}^i)\boldsymbol{V}_{0i}^i \tag{2.80}$$

$$\boldsymbol{M}_{i,RB} = \begin{bmatrix} m\boldsymbol{I}_{3\times 3} & -m\widehat{\boldsymbol{r}}_{i,ci}^i \\ m\widehat{\boldsymbol{r}}_{i,ci}^i & \boldsymbol{I}_g \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} \end{bmatrix} \tag{2.81}$$

$$\boldsymbol{C}_{i,RB} = \begin{bmatrix} \boldsymbol{0}_{3\times 3} & -\widehat{\boldsymbol{M}_{11}\boldsymbol{v}_i} - \widehat{\boldsymbol{M}_{12}\boldsymbol{\omega}_i} \\ -\widehat{\boldsymbol{M}_{11}\boldsymbol{v}_i} - \widehat{\boldsymbol{M}_{12}\boldsymbol{\omega}_i} & -\widehat{\boldsymbol{M}_{21}\boldsymbol{v}_i} - \widehat{\boldsymbol{M}_{22}\boldsymbol{\omega}_i} \end{bmatrix} \tag{2.82}$$

Using the Jacobian as in (2.73) we get the inertial forces acting on body i, projected on the generalized forces $\boldsymbol{\tau}$

$$\boldsymbol{\tau}_i = (\boldsymbol{J}_{i0})^\top \boldsymbol{M}_{i,RB}\left(\dot{\boldsymbol{J}}_{gi}^B\boldsymbol{\zeta} + \boldsymbol{J}_{i0}\dot{\boldsymbol{\zeta}}\right) + (\boldsymbol{J}_{i0})^\top \boldsymbol{C}_{i,RB}(\boldsymbol{V}_{0i}^i)\boldsymbol{J}_{i0}\boldsymbol{\zeta} \tag{2.83}$$

Summing over $i = [b, n]$ to get the contribution of the inertial forces from all bodies yields

$$\boldsymbol{\tau} = \sum_{i=b}^{n}(\boldsymbol{J}_{i0})^\top \boldsymbol{M}_{i,RB}\boldsymbol{J}_{i0}\dot{\boldsymbol{\zeta}} + (\boldsymbol{J}_{i0})^\top \boldsymbol{M}_{i,RB}\dot{\boldsymbol{J}}_{gi}^B\boldsymbol{\zeta} + (\boldsymbol{J}_{i0})^\top \boldsymbol{C}_{i,RB}(\boldsymbol{V}_{0i}^i)\boldsymbol{J}_{i0}\boldsymbol{\zeta} \tag{2.84}$$

$$= \boldsymbol{M}_{RB}(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}_{RB}(\boldsymbol{q},\boldsymbol{\zeta})\boldsymbol{\zeta} \tag{2.85}$$

where (2.86)

$$\boldsymbol{M}_{RB}(\boldsymbol{q}) := \sum_{i=b}^{n}(\boldsymbol{J}_{i0})^\top \boldsymbol{M}_{i,RB}\boldsymbol{J}_{i0} \tag{2.87}$$

$$\boldsymbol{C}_{RB}(\boldsymbol{q},\boldsymbol{\zeta}) := \sum_{i=b}^{n}(\boldsymbol{J}_{i0})^\top \boldsymbol{M}_{i,RB}\dot{\boldsymbol{J}}_{gi}^B\boldsymbol{\zeta} + (\boldsymbol{J}_{i0})^\top \boldsymbol{C}_{i,RB}(\boldsymbol{V}_{0i}^i)\boldsymbol{J}_{i0} \tag{2.88}$$

The same equations are presented in From et al. [2013]:

$$\boldsymbol{M}_{RB}(\boldsymbol{q}) = \sum_{i=b}^{n}(\boldsymbol{J}_{i0})^\top(\boldsymbol{q})\boldsymbol{I}_i\boldsymbol{J}_{i0}(\boldsymbol{q}) \tag{2.89}$$

$$\boldsymbol{C}_{RB}(\boldsymbol{q},\boldsymbol{\zeta}) = \sum_{i=b}^{n}\left((\boldsymbol{J}_{i0})^\top(\boldsymbol{q})\boldsymbol{I}_i\dot{\boldsymbol{J}}_{0i}(\boldsymbol{q}) - (\boldsymbol{J}_{i0})^\top(\boldsymbol{q})\boldsymbol{W}_i(\boldsymbol{V}_{0i}^B)\boldsymbol{J}_{i0}\right) \tag{2.90}$$

where

$$\boldsymbol{W}_i(\boldsymbol{V}_{0i}^B) = \begin{bmatrix} 0 & \widehat{\frac{\partial \mathcal{K}_i}{\partial \boldsymbol{v}_{0i}^B}} \\ \widehat{\frac{\partial \mathcal{K}_i}{\partial \boldsymbol{v}_{0i}^B}} & \widehat{\frac{\partial \mathcal{K}_i}{\partial \boldsymbol{\omega}_{0i}^B}} \end{bmatrix} \tag{2.91}$$

where $\boldsymbol{I}_s = \boldsymbol{M}_{i,RB}$ is the inertia matrix of body i. To use the system velocity $\boldsymbol{\zeta}$ it should be noted that the linear and angular velocities of each body is found through the geometrical jacobian $\boldsymbol{J}_{i0}$ in the usual way. The time derivative of the jacobian $\dot{\boldsymbol{J}}_{gi}^B$ and

$\boldsymbol{W}_i(\boldsymbol{V}_{0i}^B)$ is written in the appendix. One can see that the equations in (2.84) is the same as the equations in (2.89) and (2.91) where $\boldsymbol{W}_i = -\boldsymbol{C}_{i,RB}$ and $\boldsymbol{I}_i = \boldsymbol{M}_{i,RB}$.

From From et al. [2013] we get that the mass matrix $\boldsymbol{M}_{RB}$ of a general vehicle-manipulator system has the following properties: The inertia matrix $\boldsymbol{M}_{RB}(\boldsymbol{q})$ is uniformly bounded by d_1 and d_2 if there are no singularities present in the formulation.

$$0 < d_1 \leq ||\boldsymbol{M}_{RB}(\boldsymbol{q})|| \leq d_2 < \infty \ , \ \forall \boldsymbol{q} \in \mathbb{R}^n \tag{2.92}$$

where $||\cdot||$ is the induced norm for matrices. Since the mass matrix of the UVMS only is dependent on the joint angles, there are no singularities present in the formulation, and thus, the boundedness property in (2.92) holds. In From et al. [2013] the following are also shown:

$$(\dot{\boldsymbol{M}}_{RB} - 2\boldsymbol{C}_{RB})^\top = -(\dot{\boldsymbol{M}}_{RB} - 2\boldsymbol{C}_{RB}) \tag{2.93}$$

which shows the skew-symmetry of $(\dot{\boldsymbol{M}}_{RB} - 2\boldsymbol{C}_{RB})$ frequently used in control design, especially when proving Lyapunov stability.

Remark 2. *The matrix $\boldsymbol{M}_{RB}$ in* (2.87) *is positive definite.*

Proof. The constant mass matrix of a single rigid body (here denoted $\boldsymbol{M}_{i,RB}$) is positive definite (see e.g. Spong and Hutchinson [2005]), therefore, by definition, the following holds:

$$\boldsymbol{x}^\top \boldsymbol{M}_{i,RB} \boldsymbol{x} > 0, \ \forall \boldsymbol{x} \in \mathbb{R}^n / \{0\} \tag{2.94}$$

We now substitute $\boldsymbol{x}$ with $\boldsymbol{x} = \boldsymbol{J}_{i0}\boldsymbol{y}$, where $\boldsymbol{y}$ is any vector in $\mathbb{R}^{12}$. $\boldsymbol{J}_{i0}$ is always defined and non-singular, therefore, by substituting for $\boldsymbol{x}$ in (2.94) we get :

$$\boldsymbol{y}^\top \boldsymbol{J}_{i0}^\top \boldsymbol{M}_{i,RB} \boldsymbol{J}_{i0} \boldsymbol{y} > 0, \ \forall \boldsymbol{y} \in \mathbb{R}^m / \{0\} \tag{2.95}$$

Thus, the matrices $\boldsymbol{J}_{i0}^\top \boldsymbol{M}_{i,RB} \boldsymbol{J}_{i0}$, $i \in [b, i]$ are positive definite. Since the sum of positive definite matrices are also positive definite, we get that $\boldsymbol{M}_{RB}$ is positive definite. □

2.4.3.2 Added Mass

When moving rigid bodies in water, there is a contribution to the total forces of the system that comes from accelerating and rotating the ambient water, which is referred to as *added*

mass. The added mass matrix $\boldsymbol{M}_A$ for a rigid body is symmetric and positive semi-definite (Fossen [2011])

$$\boldsymbol{M}_A = \boldsymbol{M}_A^\top \tag{2.96}$$

$$\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x} \geq 0, \ \forall \boldsymbol{x} \in \mathbb{R}^6 \tag{2.97}$$

and for a single rigid body with three planes of symmetry, the added mass matrix can be written Fossen [2011]

$$\boldsymbol{M}_{A,i} = -diag\{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\} \tag{2.98}$$

by modeling the rigid bodies of the manipulator as cylinders, the following is an estimate of the coefficients of the added mass matrix for each link in the manipulator (Schjølberg and Fossen [1994])

$$\begin{aligned}
X_{\dot{u}} &= -10\% \text{ of the body mass} \\
Y_{\dot{v}} &= -\pi\rho r^2 L \\
Z_{\dot{w}} &= -\pi\rho r^2 L \\
K_{\dot{p}} &= 0 \\
M_{\dot{q}} &= -\tfrac{1}{12}\pi\rho r^2 L \\
N_{\dot{r}} &= -\tfrac{1}{12}\pi\rho r^2 L
\end{aligned}$$

where ρ is the density of the ambient water, r and L is the radius and length of the cylinder. Fossen [2011] then states that the added coriolis matrix of a body with three planes of symmetry can be written

$$\boldsymbol{C}_A(\boldsymbol{V}_r) = -\boldsymbol{C}_A^\top(\boldsymbol{V}_r) = \begin{bmatrix}
0 & 0 & 0 & 0 & -Z_{\dot{w}_r} w_r & Y_{\dot{v}_r} v_r \\
0 & 0 & 0 & Z_{\dot{w}_r} w_r & 0 & -X_{\dot{u}_r} u_r \\
0 & 0 & 0 & -Y_{\dot{v}_r} v_r & X_{\dot{u}_r} u_r & 0 \\
0 & -Z_{\dot{w}_r} w_r & Y_{\dot{v}_r} v_r & 0 & -N_{\dot{r}_r} r_r & M_{\dot{q}_r} q_r \\
Z_{\dot{w}_r} w_r & 0 & -X_{\dot{u}_r} u_r & N_{\dot{r}_r} r_r & 0 & -K_{\dot{p}_r} p_r \\
-Y_{\dot{v}_r} v_r & X_{\dot{u}_r} u_r & 0 & -M_{\dot{q}_r} q_r & K_{\dot{p}_r} p_r & 0
\end{bmatrix} \tag{2.99}$$

where $\boldsymbol{V}_r$ is the velocity of the rigid body relative to the ambient water. The contribution from the added mass and coriolis effect can then be summed up in the same way as (2.89)

and (2.91) by changing the rigid body inertia $\boldsymbol{M}_i$ with the added mass matrix $\boldsymbol{M}_{i,a}$, and using the relative velocity $\boldsymbol{\zeta}_r$:

$$\boldsymbol{M}_A(\boldsymbol{q}) = \sum_{i=b}^{n} (\boldsymbol{J}_{i0})^\top(\boldsymbol{q}) \boldsymbol{M}_{i,a} \boldsymbol{J}_{i0}(\boldsymbol{q}) \tag{2.100}$$

$$\boldsymbol{C}_A(\boldsymbol{q}, \boldsymbol{\zeta}_r) = \sum_{i=b}^{n} \Big((\boldsymbol{J}_{i0})^\top(\boldsymbol{q}) \boldsymbol{M}_{i,a} \dot{\boldsymbol{J}}^B_{gi}(\boldsymbol{q}) + (\boldsymbol{J}_{i0})^\top(\boldsymbol{q}) \boldsymbol{C}_{i,a}(\boldsymbol{V}^B_{ci}) \boldsymbol{J}_{i0} \Big) \tag{2.101}$$

It can be shown that the added mass matrix $\boldsymbol{M}_A$ is bounded

$$0 \leq d_1 \leq ||\boldsymbol{M}_A(\boldsymbol{q})|| \leq d_2 \leq \infty, \ \forall \boldsymbol{q} \in \mathbb{R}^n \tag{2.102}$$

We will not prove this, but it follows from the proof of the boundedness of the inertia matrix of a UVMS in From et al. [2013]. Furthermore, we have the following property

Remark 3. *The matrix $\dot{\boldsymbol{M}}_A - 2\boldsymbol{C}_A$ is skew symmetric*

Proof. We prove this using an approach similar to the proof of the skew symmetry of $\dot{\boldsymbol{M}}_{RB} - 2\boldsymbol{C}_{RB}$ in From et al. [2013]:

$$(\dot{\boldsymbol{M}}_A - 2\boldsymbol{C}_A) = \frac{d}{dt}\Big(\sum_{i=b}^{n} \boldsymbol{J}_{gi}^\top \boldsymbol{M}_{i,A} \boldsymbol{J}_{gi} \Big) \tag{2.103}$$

$$- 2 \sum_{i=b}^{n} \Big((\boldsymbol{J}_{gi})^\top \boldsymbol{M}_{i,a} \dot{\boldsymbol{J}}_{gi} + (\boldsymbol{J}_{gi})^\top \boldsymbol{C}_{i,a}(\boldsymbol{V}_{ci}) \boldsymbol{J}_{gi} \Big) \tag{2.104}$$

$$= \sum_{i=b}^{n} \dot{\boldsymbol{J}}_{gi}^\top \boldsymbol{M}_{i,A} \boldsymbol{J}_{gi} - \boldsymbol{J}_{gi}^\top \boldsymbol{M}_{i,a} \dot{\boldsymbol{J}}_{gi} - 2\boldsymbol{J}_{gi}^\top \boldsymbol{C}_{i,a}(\boldsymbol{V}_{ci}) \boldsymbol{J}_{gi} \tag{2.105}$$

We have that matrix A is skew-symmetric if the following holds:

$$A = \frac{1}{2}\Big(A - A^\top \Big) \tag{2.106}$$

From Fossen [2011] we get that the matrix $\boldsymbol{C}_{i,A}$ is skew symmetric (when $\boldsymbol{C}_A$ is parametrized as in (2.99)), and we get the following equality

$$\frac{1}{2}\Big(\boldsymbol{J}_{gi}^\top \boldsymbol{C}_{i,a}(\boldsymbol{V}_{ci}) \boldsymbol{J}_{gi} - \Big(\boldsymbol{J}_{gi}^\top \boldsymbol{C}_{i,a}(\boldsymbol{V}_{ci}) \boldsymbol{J}_{gi} \Big)^\top \Big) \tag{2.107}$$

$$= \boldsymbol{J}_{gi}^\top \boldsymbol{C}_{i,a}(\boldsymbol{V}_{ci}) \boldsymbol{J}_{gi} + \boldsymbol{J}_{gi}^\top \boldsymbol{C}_{i,a}(\boldsymbol{V}_{ci}) \boldsymbol{J}_{gi} \tag{2.108}$$

$$= \boldsymbol{J}_{gi}^\top \boldsymbol{C}_{i,a}(\boldsymbol{V}_{ci}) \boldsymbol{J}_{gi} \tag{2.109}$$

which shows that $\boldsymbol{J}_{gi}^\top \boldsymbol{C}_{i,a}(\boldsymbol{V}_{ci}) \boldsymbol{J}_{gi}$ is skew-symmetric. Furthermore, since $\boldsymbol{M}_A = \boldsymbol{M}_A^\top$ we get that

$$\left(\dot{\boldsymbol{J}}_{gi}^{\top}\boldsymbol{M}_{i,A}\boldsymbol{J}_{gi}-\boldsymbol{J}_{gi}^{\top}\boldsymbol{M}_{i,a}\dot{\boldsymbol{J}}_{gi}\right)^{\top}=-\dot{\boldsymbol{J}}_{gi}^{\top}\boldsymbol{M}_{i,A}\boldsymbol{J}_{gi}-\boldsymbol{J}_{gi}^{\top}\boldsymbol{M}_{i,a}\dot{\boldsymbol{J}}_{gi} \tag{2.110}$$

which shows the skew-symmetry of $\dot{\boldsymbol{J}}_{gi}^{\top}\boldsymbol{M}_{i,A}\boldsymbol{J}_{gi}-\boldsymbol{J}_{gi}^{\top}\boldsymbol{M}_{i,a}\dot{\boldsymbol{J}}_{gi}$. We then use that the sum of skew-symmetric matrices is itself skew-symmetric and can therefore conclude that $\dot{\boldsymbol{M}}_A - 2\boldsymbol{C}_A$ is skew-symmetric. □

2.4.3.3 Potential Forces

The potential forces consists of the gravitational force and the buoyancy acting on each body of the UVMS. The forces of gravity acting in CG of a link can be written as a wrench:

$$\boldsymbol{F}_{i,CG}=\begin{bmatrix}-m_i g\boldsymbol{R}_{i0}e_z\\ \boldsymbol{0}_{3\times 1}\end{bmatrix} \tag{2.111}$$

and the forces of buoyancy acting in CB can be written:

$$\boldsymbol{F}_{i,CB}=\begin{bmatrix}\rho\nabla_i\boldsymbol{R}_{i0}e_z\\ \boldsymbol{0}_{3\times 1}\end{bmatrix} \tag{2.112}$$

Where $e_z=\begin{bmatrix}0 & 0 & 1\end{bmatrix}^{\top}$, m_i is the mass of body i, ∇_i is the volume of body i, and ρ is the density of the surrounding water. By using the transformations in (2.78) and (2.79) we can sum the two together and express them in $\mathcal{F}_i$:

$$\boldsymbol{F}_i=\boldsymbol{Ad}_{g_{ici}}^{-\top}\begin{bmatrix}-m_i g\boldsymbol{R}_{i0}e_z\\ \boldsymbol{0}_{3\times 1}\end{bmatrix}+\boldsymbol{Ad}_{g_{ibi}}^{-\top}\begin{bmatrix}\rho\nabla_i\boldsymbol{R}_{i0}e_z\\ \boldsymbol{0}_{3\times 1}\end{bmatrix} \tag{2.113}$$

We then get that the contribution from the potential forces can be summed up by the contribution of each body i

$$\boldsymbol{N}(\boldsymbol{\xi})=\sum_{i=b}^{n}(\boldsymbol{J}_{i0})^{\top}\left(\boldsymbol{Ad}_{g_{ici}}^{-\top}\begin{bmatrix}-m_i g\boldsymbol{R}_{i0}e_z\\ \boldsymbol{0}_{3\times 1}\end{bmatrix}+\boldsymbol{Ad}_{g_{ibi}}^{-\top}\begin{bmatrix}\rho\nabla_i\boldsymbol{R}_{i0}e_z\\ \boldsymbol{0}_{3\times 1}\end{bmatrix}\right) \tag{2.114}$$

2.4.3.4 Hydrodynamics

The hydrodynamical forces on each body are a function of the relative velocity between the body and the surrounding water. In marine craft motion control, the hydrodynamical damping on a general rigid body is often written in matrix form (see e.g. Fossen [2011])

$$\boldsymbol{\tau}=\boldsymbol{D}(\boldsymbol{V}_r)\boldsymbol{V}_r \tag{2.115}$$

Where $\boldsymbol{V}_r$ is the velocity of the rigid body relative to the surrounding water. Furthermore, for a rigid body moving through an ideal fluid, the hydrodynamic matrix is positive definite (Fossen [2011])

$$\boldsymbol{x}^\top \boldsymbol{D}(\boldsymbol{V})\boldsymbol{x} > 0,\ \forall \boldsymbol{x} \in \mathbb{R}^6/\{0\} \tag{2.116}$$

We now associate a matrix $\boldsymbol{D}_i(\boldsymbol{V}_{0i})$ with each of the bodies of the UVMS. An in-depth discussion of the hydrodynamics of each rigid body is outside the scope of this text, and we neglect higher order terms, keeping only the second and first order terms in $\boldsymbol{V}_{0i}$. From Antonelli [2013] we get that the following approximation can be used for a single rigid body

$$\boldsymbol{D}_i(\boldsymbol{V}_{0i}) = -\, diag\{X_u, Y_v, Z_w, K_p, M_q, N_r\} \tag{2.117}$$

$$-\, diag\{X_{u|u|}|u|, Y_{v|v|}|v|, Z_{w|w|}|w|, K_{p|p|}|p|, M_{q|q|}|q|, N_{r|r|}|r|\} \tag{2.118}$$

where the velocities u, v, w, u, q, r are the different components of the $\boldsymbol{V}_{0i}$. Since $\boldsymbol{D}_i$ is calculated in CG of each body we can write the wrench expressed in CG as a function of the velocity of CG:

$$\boldsymbol{F}_{ci} = \boldsymbol{D}_i(\boldsymbol{V}_{ci,r})\boldsymbol{V}_{ci,r} \tag{2.119}$$

Where $\boldsymbol{V}_{ci,r}$ is the velocity CG of link i relative to the current. We need to map the general velocities to the velocities of CG, and also write the wrench in the coordinates of $\mathcal{F}_i$.

$$\boldsymbol{V}_{ci,r} = \boldsymbol{Ad}_{g_{ici}}^{-1}\boldsymbol{J}_{i0}\boldsymbol{\zeta}_r \tag{2.120}$$

$$\boldsymbol{F}_i = \boldsymbol{Ad}_{g_{ici}}^{-\top}\boldsymbol{F}_{ci} \tag{2.121}$$

and insert it into (2.119) to finally obtain the total hydrodynamical forces of the system

$$\begin{aligned}\boldsymbol{\tau} &= \sum_{i=b}^{n}(\boldsymbol{J}_{i0})^\top \boldsymbol{Ad}_{g_{ici}}^{-\top}\boldsymbol{D}_i\left(\boldsymbol{Ad}_{g_{ici}}^{-1}\boldsymbol{J}_{i0}\boldsymbol{\zeta}_r\right)\boldsymbol{Ad}_{g_{ici}}^{-1}\boldsymbol{J}_{i0}\boldsymbol{\zeta}_r \\ &= \boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta}_r)\boldsymbol{\zeta}_r\end{aligned} \tag{2.122}$$

where

$$\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta}_r) := \sum_{i=b}^{n}(\boldsymbol{J}_{i0})^\top \boldsymbol{Ad}_{g_{ici}}^{-\top}\boldsymbol{D}_i\left(\boldsymbol{Ad}_{g_{ici}}^{-1}\boldsymbol{J}_{i0}\boldsymbol{\zeta}_r\right)\boldsymbol{Ad}_{g_{ici}}^{-1}\boldsymbol{J}_{i0}$$

Remark 4. *The matrix $\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\zeta}_r)$ is positive definite.*

Proof. The hydrodynamical matrix of a single rigid body $\boldsymbol{D}_i$ is positive definite Fossen [2011] and therefore

$$\boldsymbol{x}^\top \boldsymbol{D}_i \boldsymbol{x} > 0, \ \forall \boldsymbol{x} \in \mathbb{R}^n / \{0\} \tag{2.123}$$

we now substitute $\boldsymbol{x}$ with $\boldsymbol{x} = \boldsymbol{Ad}_{g_{ici}}^{-1} \boldsymbol{J}_{i0} \boldsymbol{y}$, for any vector $\boldsymbol{y}$ in $\mathbb{R}^{12}$. $\boldsymbol{Ad}_{g_{ici}}^{-1} \boldsymbol{J}_{i0}$ is always defined and non-singular, therefore, by substituting for $\boldsymbol{x}$ in (2.123) we get :

$$\boldsymbol{y}^\top \boldsymbol{J}_{i0}^\top \boldsymbol{Ad}_{g_{ici}}^{-\top} \boldsymbol{D}_i \boldsymbol{Ad}_{g_{ici}}^{-1} \boldsymbol{J}_{i0} \boldsymbol{y} > 0, \ \forall \boldsymbol{y} \in \mathbb{R}^m / \{0\} \tag{2.124}$$

Thus, the matrices $\boldsymbol{J}_{i0}^\top \boldsymbol{Ad}_{g_{ici}}^{-\top} \boldsymbol{D}_i \boldsymbol{Ad}_{g_{ici}}^{-1} \boldsymbol{J}_{i0} \quad i \in [b, i]$ are positive definite. Since the sum of positive definite matrices are also positive definite, we get that $\boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\zeta}_r)$ is positive definite. □

2.4.3.5 Total Dynamics Equation

Using the above matrices, the total dynamics of the UVMS can be expressed by

$$\boldsymbol{M}_{RB}(\boldsymbol{\xi})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}_{RB}(\boldsymbol{\xi}, \boldsymbol{\zeta})\boldsymbol{\zeta} + \boldsymbol{M}_A(\boldsymbol{\xi})\dot{\boldsymbol{\zeta}}_r + \boldsymbol{C}_A(\boldsymbol{\xi}, \boldsymbol{\zeta}_r)\boldsymbol{\zeta}_r + \boldsymbol{D}(\boldsymbol{\xi}, \boldsymbol{\zeta}_r)\boldsymbol{\zeta}_r + \boldsymbol{N}(\boldsymbol{\xi}) = \boldsymbol{\tau}_c \tag{2.125}$$

where $\boldsymbol{\tau}_c$ is the commanded torque output from the controller. From (2.57) and (2.59) we have that $\dot{\boldsymbol{\zeta}}_r = \dot{\boldsymbol{\zeta}} - \boldsymbol{H}_n \boldsymbol{v}_c^0$, and $\boldsymbol{\zeta}_r = \boldsymbol{\zeta} - \boldsymbol{H}_m \boldsymbol{v}_c^0$. We can then write (2.125) in a more compact way,

$$\boldsymbol{M}(\boldsymbol{\xi})\dot{\boldsymbol{\zeta}} + \boldsymbol{C}(\boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{\zeta}_r)\boldsymbol{\zeta} + \boldsymbol{D}(\boldsymbol{\xi}, \boldsymbol{\zeta}_r)\boldsymbol{\zeta} + \boldsymbol{N}(\boldsymbol{\xi}) + \boldsymbol{\Xi}(\boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{\zeta}_r) = \boldsymbol{\tau}_c \tag{2.126}$$

Where

$$\boldsymbol{M}(\boldsymbol{\xi}) := \boldsymbol{M}_{RB}(\boldsymbol{\xi}) + \boldsymbol{M}_A(\boldsymbol{\xi}) \tag{2.127}$$

$$\boldsymbol{C}(\boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{\zeta}_r) := \boldsymbol{C}_{RB}(\boldsymbol{\xi}, \boldsymbol{\zeta}) + \boldsymbol{C}_A(\boldsymbol{\xi}, \boldsymbol{\zeta}_r) \tag{2.128}$$

$$\boldsymbol{\Xi}(\boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{\zeta}_r) := -\boldsymbol{M}_A(\boldsymbol{\xi})\boldsymbol{H}_n(\boldsymbol{\xi}, \boldsymbol{\zeta})\boldsymbol{v}_c^0 - \Big(\boldsymbol{C}_A(\boldsymbol{\xi}, \boldsymbol{\zeta}_r) + \boldsymbol{D}(\boldsymbol{\xi}, \boldsymbol{\zeta}_r)\Big)\boldsymbol{H}_m(\boldsymbol{\xi})\boldsymbol{v}_c^0 \tag{2.129}$$

and $\boldsymbol{v}_c^0$ is the velocity of the ocean current in $\mathcal{F}_0$.

Remark 5. *The inertia matrix $\boldsymbol{M}(\boldsymbol{\xi})$ in (2.126) is positive definite and bounded*

This follows from properties of $\boldsymbol{M}_{RB}$ and $\boldsymbol{M}_A$ which is both bounded. $\boldsymbol{M}_{RB}$ is positive definite, while $\boldsymbol{M}_A$ is positive semi definite. Consequently the sum of the two must be bounded, and positive definite.

Remark 6. *The matrix $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$ is skew symmetric.*

Proof. We have that

$$\dot{\boldsymbol{M}}_{RB} - 2\boldsymbol{C}_{RB} = -(\dot{\boldsymbol{M}}_{RB} - 2\boldsymbol{C}_A)^\top \tag{2.130}$$

$$\dot{\boldsymbol{M}}_A - 2\boldsymbol{C}_A = -(\dot{\boldsymbol{M}}_A - 2\boldsymbol{C}_A)^\top \tag{2.131}$$

and we get that

$$\dot{\boldsymbol{M}} - 2\boldsymbol{C} = \dot{\boldsymbol{M}}_{RB} - 2\boldsymbol{C}_{RB} + \dot{\boldsymbol{M}}_A - 2\boldsymbol{C}_A \tag{2.132}$$

$$= -(\dot{\boldsymbol{M}}_{RB} - 2\boldsymbol{C}_A)^\top - (\dot{\boldsymbol{M}}_A - 2\boldsymbol{C}_A)^\top \tag{2.133}$$

$$= -(\dot{\boldsymbol{M}} - 2\boldsymbol{C})^\top \tag{2.134}$$

and $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$ is therefore skew symmetric. □

The formulation in (2.126) is different from formulations in other literature on underwater vehicle-manipulator systems, as it writes matrices which are dependent on both the system velocity $\boldsymbol{\zeta}$ and the relative velocity $\boldsymbol{\zeta}_r$. We also include a term $\boldsymbol{\Xi}$ that is only nonzero for a nonzero $\boldsymbol{v}_c^0$. Although this yields terms that are not as intuitively meaningful as in other formulations, they still have the properties of boundedness, skew symmetry and positive definiteness, which are important in control system design.

3 | Control

The control scheme presented in this chapter consists of two main parts: an inner loop sliding mode controller (SMC), and an outer loop kinematic control system. An illustration of the total system is listed in Fig. 3.1. The reference signal to the total system is then the desired velocity of the end effector. The desired velocity will then be the output of an operator system controlling the UVMS from a surface vessel, or from a land based facility.

The kinematic control takes the desired end effector velocity, and the measured configuration of the system, and gives a desired configuration trajectory based on several objectives that are described below. The sliding mode controller takes the desired configuration trajectory $\boldsymbol{\xi}_d$ and outputs the forces $\boldsymbol{\tau}_c$ to the UVMS in order to obtain the desired configurations.

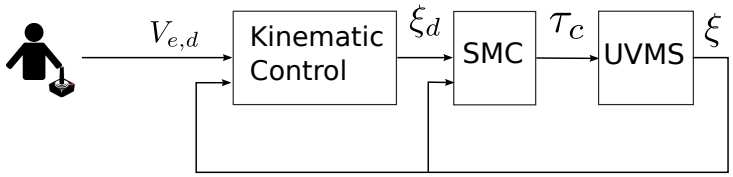


Figure 3.1: Overview of the total control-plant system

3.1 Sliding Mode Control

Sliding Mode Control (SMC) is a well-known control method which, has been applied to robotics, aircrafts and marine crafts. SMC is robust when it comes to uncertainties in model parameters as long as the errors in the parameters are bounded. This makes it ideal for control of a UVMS, where it is difficult to obtain accurate estimates for the model parameters. Antonelli [2013] presents an SMC law for controlling the configuration of a UVMS to constant desired configurations and without taking the ocean current into

account. Fossen [2011] gives an outline of an SMC law for an underwater vehicle where the current is taken into account, but this is only for SISO control, and does not include the total dynamics of the vehicle. Schjølberg and Fossen [1994] present a control method for a UVMS based on feedback linearization. This formulation requires that all the dynamics parameters are known, and does not take the ocean current into account.

To make the UVMS robust to both parameter uncertainties and ocean current, we have designed an SMC to track time varying, continuous trajectory in the coordinate space of $\boldsymbol{\xi}$. The SMC law is inspired by the singularity free tracking of an AUV in Fjellstad and Fossen [1994c]. It is, however, extended to use in a sliding mode control scheme, and includes the manipulator coupled dynamics, uncertainties in the dynamics parameters and the influence from the ocean current.
For the Sliding Mode Controller, the following manifold is used

$$\boldsymbol{s} = \boldsymbol{\zeta} - \boldsymbol{\zeta}_s \ , \ \boldsymbol{s} \in \mathbb{R}^{12} \tag{3.1}$$

where $\boldsymbol{\zeta}_s$ is a virtual velocity reference signal to be defined later. The object of the SMC is then to force the system to reach and stay on the manifold $\boldsymbol{s} = 0$, where we will show that the state error converges to zero asymptotically.

3.1.1 Stability On the Manifold $\boldsymbol{s} = \mathbf{0}$

We now define the virtual reference signal $\boldsymbol{\zeta}_s$:

$$\boldsymbol{\zeta}_s := \boldsymbol{\zeta}_d + \boldsymbol{\Lambda e} \tag{3.2}$$

Where $\boldsymbol{\zeta}_d = \begin{bmatrix} \boldsymbol{v}_d^\top & \boldsymbol{\omega}_d^\top & \dot{\boldsymbol{q}}_d^\top \end{bmatrix}^\top$ is the desired system velocity, and is continuously differentiable, and where

$$\boldsymbol{\Lambda} := \begin{bmatrix} \boldsymbol{K}_p & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times6} \\ \boldsymbol{0}_{3\times3} & -2c\frac{\partial W}{\partial \widetilde{\eta}}\boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times6} \\ \boldsymbol{0}_{6\times3} & \boldsymbol{0}_{6\times3} & \boldsymbol{K}_q \end{bmatrix} \in \mathbb{R}^{12\times12} \tag{3.3}$$

$$\boldsymbol{e} := \begin{bmatrix} \widetilde{\boldsymbol{x}} \\ \widetilde{\boldsymbol{\epsilon}} \\ \widetilde{\boldsymbol{q}} \end{bmatrix} \in \mathbb{R}^{12} \tag{3.4}$$

$\widetilde{\boldsymbol{x}} := \boldsymbol{x}_d - \boldsymbol{x}$ is the vehicle position error, $\widetilde{\boldsymbol{\epsilon}}$ is the error in the rotation of the vehicle as defined in (2.7) and $\widetilde{\boldsymbol{q}} := \boldsymbol{q}_d - \boldsymbol{q}$ is the manipulator joint errors. The scalar function $W(\widetilde{\eta})$ is

the same as in Fjellstad and Fossen [1994c] and is non-negative on the interval $\widetilde{\eta} \in [-1, 1]$ and vanishes only at $\widetilde{\eta} = \pm 1$. It also satisfies the Lipschitz condition on the same interval.

Stability on $\boldsymbol{s} = 0$ can now be split up into the vehicle translational part, vehicle rotational part, and the manipulator part. From the equations above we get the following error dynamics of the system on $\boldsymbol{s} = 0$

$$\widetilde{\boldsymbol{v}} = -\boldsymbol{K}_p \widetilde{\boldsymbol{x}} \tag{3.5}$$

$$\widetilde{\boldsymbol{\omega}} = 2c \frac{\partial W}{\partial \widetilde{\eta}} \widetilde{\boldsymbol{\epsilon}} \tag{3.6}$$

$$\dot{\widetilde{\boldsymbol{q}}} = -\boldsymbol{K}_q \widetilde{\boldsymbol{q}} \tag{3.7}$$

Stability on the manifold is then ensured by ensuring that the errors $\widetilde{\boldsymbol{x}}$, $\widetilde{\boldsymbol{\epsilon}}$ and $\widetilde{\boldsymbol{q}}$ converges to zero.

Manipulator Error Dynamics on the Manifold

The error convergence of $\widetilde{\boldsymbol{q}}$ can be ensured by taking $\boldsymbol{K}_q$ to be positive definite, i.e. $\boldsymbol{K}_q > O$. We choose the following Lyapunov function candidate

$$V = \frac{1}{2} \widetilde{\boldsymbol{q}}^\top \widetilde{\boldsymbol{q}} > 0 \ , \forall \widetilde{\boldsymbol{q}} \in \mathbb{R}^6 / \{0\} \tag{3.8}$$

Differentiating V yields

$$\dot{V} = \widetilde{\boldsymbol{q}}^\top \dot{\widetilde{\boldsymbol{q}}} \tag{3.9}$$

$$= -\widetilde{\boldsymbol{q}}^\top \boldsymbol{K}_q \widetilde{\boldsymbol{q}} < 0 \ , \forall \widetilde{\boldsymbol{q}} \in \mathbb{R}^6 / \{0\} \tag{3.10}$$

By application of Lyapunov's direct method we see that the equilibrium $\widetilde{\boldsymbol{q}} = \boldsymbol{0}$ is asymptotically stable.

Vehicle Rotational Error Dynamics on the Manifold

Here we use the same arguments as in Fjellstad and Fossen [1994a] to show that the rotational error converges to zero. Let $W(\widetilde{\eta})$ be a Lyapunov function candidate. Differentiation of $W(\widetilde{\eta})$ on $\boldsymbol{s} = \boldsymbol{0}$ yields

$$\dot{W}(\widetilde{\eta}) = \frac{\partial W}{\partial \widetilde{\eta}} \dot{\widetilde{\eta}} \tag{3.11}$$

$$= -\frac{1}{2} \frac{\partial W}{\partial \widetilde{\eta}} \widetilde{\boldsymbol{\epsilon}}^\top \widetilde{\boldsymbol{\omega}} \tag{3.12}$$

$$= -c \left(\frac{\partial W}{\partial \widetilde{\eta}} \right)^2 \widetilde{\boldsymbol{\epsilon}}^\top \widetilde{\boldsymbol{\epsilon}} \tag{3.13}$$

We then choose the function $W(\widetilde{\eta})$ to be

$$W(\widetilde{\eta}) = 1 - |\widetilde{\eta}| \tag{3.14}$$

which yields

$$\frac{\partial W}{\partial \widetilde{\eta}} = -\text{sign}(\widetilde{\eta}) \tag{3.15}$$

$\widetilde{\eta} = cos(\beta/2)$ and thus $|\widetilde{\eta}| \leq 1$. It then follows that the Lyapunov function candidate $W(\widetilde{\eta})$ is positive except at the equilibriums $\widetilde{\eta} = \pm 1$. The differentiated function $\dot{W}(\widetilde{\eta})$ is negative for all $\widetilde{\eta} \neq \pm 1$ and thus, $\widetilde{\eta}$ converges asymptotically to the stable equilibrium ± 1. (see Fjellstad and Fossen [1994a] for details).

Vehicle Translational Error Dynamics on the Manifold

Firstly, we define $\dot{\boldsymbol{x}}_d$ to be

$$\dot{\boldsymbol{x}}_d := \boldsymbol{R}(\boldsymbol{Q})\boldsymbol{v}_d \tag{3.16}$$

which gives the equality

$$\widetilde{\boldsymbol{v}} = \boldsymbol{R}(\boldsymbol{Q})^\top \dot{\widetilde{\boldsymbol{x}}} \tag{3.17}$$

And we define $\boldsymbol{K}_p$ to be (Fjellstad and Fossen [1994b])

$$\boldsymbol{K}_p := \lambda \boldsymbol{R}(\boldsymbol{Q}_d)^\top \ , \ \lambda > 0 \tag{3.18}$$

By substituting (3.18) and (3.17) into (3.5) we get the following error dynamics for the translational motion of the vehicle on the manifold:

$$\dot{\widetilde{\boldsymbol{x}}} = -\lambda \widetilde{\boldsymbol{R}} \widetilde{\boldsymbol{x}} \tag{3.19}$$

where

$$\widetilde{\boldsymbol{R}} := \boldsymbol{R}(\boldsymbol{Q})\boldsymbol{R}(\boldsymbol{Q}_d)^\top \tag{3.20}$$

We have that $eig(\widetilde{\boldsymbol{R}}) \in \{1, 2\widetilde{\eta}^2 - 1 \pm j2\widetilde{\eta}\sqrt{1-\widetilde{\eta}^2}\}$ (Fjellstad and Fossen [1994a]) and thus, $\widetilde{\boldsymbol{R}}$ is strictly positive for $\widetilde{\eta}^2 > 1/2$. Since we have that

$$\widetilde{\eta} = cos(\frac{\widetilde{\beta}}{2}) \tag{3.21}$$

we get that $\widetilde{\boldsymbol{R}}$ is strictly positive when $|\widetilde{\beta}| < \frac{\pi}{2}$. Since the rotation error converges uniformly to zero, the positive-definiteness of $\widetilde{\boldsymbol{R}}$ will be satisfied after, at least, a transient

period, and from the error dynamics in (3.19) it follows that $\widetilde{\boldsymbol{x}}$ converges to zero.

Finally, from the discussion above we can write the virtual velocity reference $\boldsymbol{\zeta}_s$ as:

$$\boldsymbol{\zeta}_s = \begin{bmatrix} \boldsymbol{v}_d + \lambda \boldsymbol{R}(\boldsymbol{Q}_d)^\top \widetilde{\boldsymbol{x}} \\ \boldsymbol{\omega}_d + 2c \ \text{sign}(\widetilde{\eta}) \widetilde{\boldsymbol{\epsilon}} \\ \dot{\boldsymbol{q}}_d + \boldsymbol{K}_q \widetilde{\boldsymbol{q}} \end{bmatrix} \tag{3.22}$$

3.1.2 Convergence To the Manifold $\boldsymbol{s} = \boldsymbol{0}$

To ensure that the system converges to the sliding manifold $\boldsymbol{s} = 0$, we use the following Lyapunov function candidate

$$V = \frac{1}{2} \boldsymbol{s}^\top \boldsymbol{M} \boldsymbol{s} > 0, \ \forall \boldsymbol{s} \in \mathbb{R}^{12} / \{0\} \tag{3.23}$$

Taking the time derivative of V along the trajectory of the system and substituting (2.126) into (3.1) yields

$$\begin{aligned} \dot{V} =& \boldsymbol{s}^\top \boldsymbol{M} \dot{\boldsymbol{s}} + \frac{1}{2} \boldsymbol{s}^\top \dot{\boldsymbol{M}} \boldsymbol{s} \\ =& \boldsymbol{s}^\top \boldsymbol{M} \left(\dot{\boldsymbol{\zeta}} - \dot{\boldsymbol{\zeta}}_s \right) + \frac{1}{2} \boldsymbol{s}^\top \dot{\boldsymbol{M}} \boldsymbol{s} \\ =& \boldsymbol{s}^\top \left(\boldsymbol{\tau}_c - \boldsymbol{C}\boldsymbol{\zeta} - \boldsymbol{D}\boldsymbol{\zeta} - \boldsymbol{N} - \boldsymbol{\Xi} - \boldsymbol{M}\dot{\boldsymbol{\zeta}}_s \right) + \frac{1}{2} \boldsymbol{s}^\top \dot{\boldsymbol{M}} \boldsymbol{s} \end{aligned} \tag{3.24}$$

We then use that

$$\boldsymbol{\zeta} = \boldsymbol{s} + \boldsymbol{\zeta}_s \tag{3.25}$$

and obtain

$$\dot{V} = \boldsymbol{s}^\top \left(\boldsymbol{\tau}_c - \boldsymbol{C}\boldsymbol{\zeta}_s - \boldsymbol{D}\boldsymbol{\zeta}_s - \boldsymbol{N} - \boldsymbol{\Xi} - \boldsymbol{M}\dot{\boldsymbol{\zeta}}_s \right) + \frac{1}{2} \boldsymbol{s}^\top (\dot{\boldsymbol{M}} - 2\boldsymbol{C}) \boldsymbol{s} - \boldsymbol{s}^\top \boldsymbol{D} \boldsymbol{s} \tag{3.26}$$

Since the matrix $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$ is skew symmetric, $\frac{1}{2} \boldsymbol{s}^\top (\dot{\boldsymbol{M}} - 2\boldsymbol{C}) \boldsymbol{s} = 0$ and we obtain

$$\dot{V} = \boldsymbol{s}^\top (\boldsymbol{\tau}_c - \boldsymbol{C}\boldsymbol{\zeta}_s - \boldsymbol{D}\boldsymbol{\zeta}_s - \boldsymbol{N} - \boldsymbol{\Xi} - \boldsymbol{M}\dot{\boldsymbol{\zeta}}_s) - \boldsymbol{s}^\top \boldsymbol{D} \boldsymbol{s} \tag{3.27}$$

We now propose the following control input

$$\boldsymbol{\tau}_c = \hat{\boldsymbol{M}} \dot{\boldsymbol{\zeta}}_s + \hat{\boldsymbol{C}} \boldsymbol{\zeta}_s + \hat{\boldsymbol{D}} \boldsymbol{\zeta}_s + \hat{\boldsymbol{N}} + \hat{\boldsymbol{\Xi}} - \boldsymbol{K}_d \boldsymbol{s} - \boldsymbol{K}_s sign(\boldsymbol{s}) \tag{3.28}$$

where $\hat{\cdot}$ denotes the estimate of a matrix or vector, and where $sign(\cdot)$ maps a vector in $\mathbb{R}^n$ to a vector in $\mathbb{R}^n$, and is defined as

$$sign_i(x_i) = \begin{cases} 1 & \text{if } x_i > 0 \\ 0 & \text{if } x_i = 0 \\ -1 & \text{otherwise} \end{cases} \tag{3.29}$$

Substituting (3.28) into (3.27) then yields

$$\dot{V} = -\boldsymbol{s}^\top (\boldsymbol{D} + \boldsymbol{K}_d)\, \boldsymbol{s} - \boldsymbol{s}^\top \left(\widetilde{\boldsymbol{M}}\dot{\boldsymbol{\zeta}}_s + \widetilde{\boldsymbol{C}}\boldsymbol{\zeta}_s + \widetilde{\boldsymbol{D}}\boldsymbol{\zeta}_s + \widetilde{\boldsymbol{N}} + \widetilde{\boldsymbol{\Xi}} + \boldsymbol{K}_s sign(\boldsymbol{s}) \right) \tag{3.30}$$

where $\tilde{\cdot}$ denotes the difference between the actual and estimated parameter values. Furthermore, we define the vector $\tilde{\boldsymbol{Y}}$

$$\tilde{\boldsymbol{Y}} = \widetilde{\boldsymbol{M}}\dot{\boldsymbol{\zeta}}_s + \widetilde{\boldsymbol{C}}\boldsymbol{\zeta}_s + \widetilde{\boldsymbol{D}}\boldsymbol{\zeta}_s + \widetilde{\boldsymbol{N}} + \widetilde{\boldsymbol{\Xi}} \tag{3.31}$$

and we get the following

$$\dot{V} = -\boldsymbol{s}^\top (\boldsymbol{D} + \boldsymbol{K}_d)\, \boldsymbol{s} - \boldsymbol{s}^\top \left(\tilde{\boldsymbol{Y}} + \boldsymbol{K}_s sign(\boldsymbol{s}) \right) \tag{3.32}$$

Since the matrix $(\boldsymbol{D} + \boldsymbol{K}_d)$ is positive definite, we can put the following upper bound on $\dot{V}$:

$$\dot{V} < -\lambda_{min}(\boldsymbol{K}_s)\|\boldsymbol{s}\| + \left\|\tilde{\boldsymbol{Y}}\right\|\|\boldsymbol{s}\| \tag{3.33}$$

where $\lambda_{min}(\boldsymbol{K}_s)$ denotes the smallest eigenvalue of $\boldsymbol{K}_s$. We then set the following criteria for choosing $\boldsymbol{K}_s$:

$$\lambda_{min}(\boldsymbol{K}_s) \geq \left\|\tilde{\boldsymbol{Y}}\right\| \tag{3.34}$$

By choosing $\boldsymbol{K}_s$ according to (3.34) we ensure that $\dot{V}$ is negative definite. Then application of Lyapunov's direct method theorem for non-autonomous systems yields globally uniformly asymptotically stability of the equilibrium $\boldsymbol{s} = \boldsymbol{0}$ (Khalil [2002]).

As customary in sliding mode control, we will use a saturation function $sat(\boldsymbol{s})$ instead of the sign function $sign(\boldsymbol{s})$ in (3.28). This is to avoid *chattering* which is caused by the discontinuity of the sign function. The sat function can be defined as

$$sat(x) = \begin{cases} sgn(x) & \text{if } |x/\epsilon| > 1 \\ x/\epsilon & \text{otherwise} \end{cases}$$

and when operating on a vector:

$$sat(\boldsymbol{y}) = \begin{bmatrix} sat(y_1) & sat(y_2) \dots & sat(y_n) \end{bmatrix}^\top , \ \boldsymbol{y} \in \mathbb{R}^n$$

where the parameter ϵ can be tuned sufficiently low. The asymptotic convergence to $\boldsymbol{s} = 0$ is then only assured for $\epsilon = 0$ (where $sat(\boldsymbol{s}) = sign(\boldsymbol{s})$). For a non-zero ϵ, however, $\boldsymbol{s}$ can only be ultimate bounded, where the bound on $\boldsymbol{s}$ can be reduced by decreasing ϵ (Khalil [2002]). Thus, for practical use, $\boldsymbol{s}$ can get sufficiently close to zero by tuning ϵ sufficiently low.

3.2 Kinematic Control

As customary in robotics control, one is interested in controlling the motion of the end effector, in order to do different tasks. The sliding mode control presented above works directly in the configuration space of $\boldsymbol{\xi}$, and we therefore need to assign the references $\boldsymbol{\xi}_d$ that corresponds to the desired end effector motion. In order to do this, we will use the velocity kinematics, with the end effector Jacobian, which maps the system velocities to the end effector twist. However, this mapping is not one-to-one because the total system has more degrees of freedom than the end effector, and the kinematics of the system is said to be *redundant*. For any given end effector trajectory we can then choose different corresponding system-velocities using inverse kinematics. We can then use the system velocity to obtain other objectives, besides only tracking the end effector. The objectives that we will obtain, while tracking the end effector trajectory, are:

- Avoid reaching the manipulator joint limits
- Keep the vehicle stationary as long as the desired end effector pose is possible to reach.

The objectives listed above are obtained through a well-known inverse kinematic technique, using the *weighted pseudo inverse* of the end effector Jacobian $\boldsymbol{J}_{e0}$. This solves the problem of manipulator joint limits. In order to obtain the second objective, we present an *Event Based Vehicle Kinematic Control* that is described below. This method yields vehicle

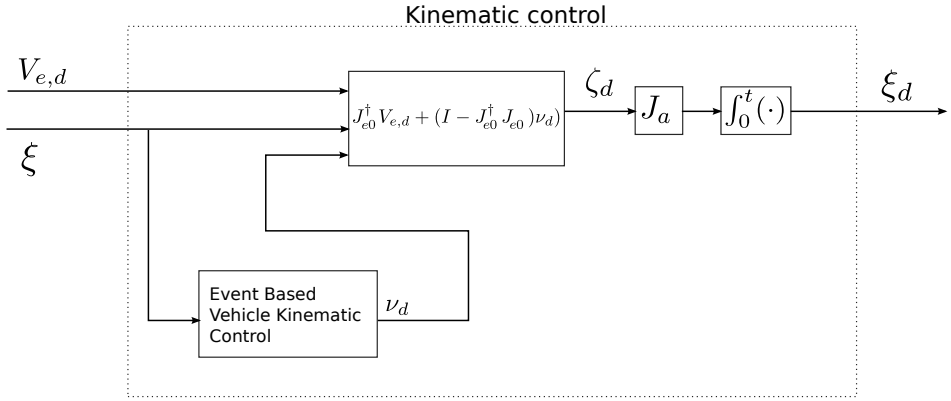


Figure 3.2: Overview of the kinematic control system

velocities in order to keep the end effector in a position relative to the vehicle, so that the manipulator is fully dexterous. In Fig. 3.2 the total kinematic control system is illustrated.

3.2.1 Weighted Least-Norm Pseudo Inverse Jacobian for Avoiding Joint Limits

The geometric Jacobian represents a transformation from the system velocity $\boldsymbol{\zeta}$ to the end effector velocity $\boldsymbol{V}_{0e}$

$$\boldsymbol{V}_{0e} = \boldsymbol{J}_{e0}\boldsymbol{\zeta} \tag{3.35}$$

In this section, however, the aim is to get the system velocity from the desired trajectory $\boldsymbol{V}_{0e}$. In kinematic control of robot manipulator, this is generally done using the pseudo inverse of $\boldsymbol{J}_{e0}$, namely $\boldsymbol{J}_{e0}^{\dagger}$.

Chan and Dubey [1995] propose a weighted Least-Norm solution to the inverse kinematic problem for avoiding joint limits. The weighted least-norm solution yields

$$\boldsymbol{J}_{e0}^{\dagger} = \boldsymbol{W}^{-1}\boldsymbol{J}_{e0}^{\top}\left(\boldsymbol{J}_{e0}\boldsymbol{W}^{-1}\boldsymbol{J}_{e0}^{\top}\right)^{-1} \tag{3.36}$$

Where $\boldsymbol{W}$ is a positive definite diagonal matrix

$$\boldsymbol{W} = \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_{12} \end{bmatrix} \in \mathbb{R}^{12\times 12} \tag{3.37}$$

From the weighted pseudo inverse we get that

$$\boldsymbol{\zeta} = \boldsymbol{J}_{e0}^{\dagger} \boldsymbol{V}_{0e} \tag{3.38}$$

Weighting the elements along the diagonal of $\boldsymbol{W}$ higher than other elements corresponds to using less velocity on the corresponding $\boldsymbol{\zeta}$ elements. As an example, using the weighting $w_i = \infty,\ i \leq 6$ and $w_i = 1,\ i > 6$ corresponds to sole manipulator motion.

For a robot manipulator on a fixed base $\boldsymbol{W}$ can be weighted to avoid joint limits by using the weighting (Chan and Dubey [1995]):

$$w_i = \begin{cases} 1 + \left| \frac{\partial H(q)}{\partial q_i} \right| & \text{if } \Delta \left| \frac{\partial H(q)}{\partial q_i} \right| \geq 0 \\ 1 & \text{otherwise} \end{cases} \tag{3.39}$$

$$\frac{\partial H(q)}{\partial q_i} = \frac{(q_{i,max} - q_{i,min})^2 (2q_i - q_{i,max} - q_{i,min})}{4 (q_{i,min} - q_i)^2 (q_i - q_{i,min})^2} \tag{3.40}$$

The function $\frac{\partial H(q)}{\partial q_i}$ is going to infinity when the joint approaches its limit. w_i therefore goes to infinity when the joint angle q_i is approaching the limit. When q_i is going away from its limit, on the other hand, $w_i = 1$, $\Delta \frac{\partial H(q)}{\partial q}$ only changes sign when $\dot{q}_i = 0$ or $\frac{\partial H(q)}{\partial q_i}$ itself is zero, thus there is no discontinuity in $\dot{q}_i$ (Chan and Dubey [1995]).

When running on a discrete system, i.e. a computer, discontinuities can still be experienced when w_i changed from a large value to 1, due to a non-zero value of the corresponding joint link. We therefore propose to changing w_i gradually to 1, by the following

$$\begin{aligned} w_i &= 1 + \alpha_h \left| \frac{\partial H(q)}{\partial q_i} \right|_{T-1} \\ \alpha_h &\in (0, 1) \end{aligned} \tag{3.41}$$

where $T-1$ indicates that $\left| \frac{\partial H(q)}{\partial q_i} \right|_{T-1}$ is from the previous iteration. α_c is then a tuning parameter for how fast w_i returns to 1 when the manipulator joint goes away from the joint limit.

We now propose the tuning law for the weighting matrix $\boldsymbol{W}$ for avoiding joint limits of the UVMS

$$w_i = k_i\ ,\ i \leq 6 \tag{3.42}$$

$$w_i = \begin{cases} 1 + \left| \frac{\partial H(q)}{\partial q_i} \right|_T & \text{if } \left| \frac{\partial H(q)}{\partial q_i} \right|_T - \left| \frac{\partial H(q)}{\partial q_i} \right|_{T-1} \geq 0 \\ 1 + \alpha_h \left| \frac{\partial H(q)}{\partial q_i} \right|_{T-1} & \text{otherwise} \end{cases} ,\ i > 6$$

Where $k_i \gg 1$ is a constant number that ensures that w_i , $i \leq 6$, corresponding to the vehicle velocity , is larger than w_i , $i > 6$, corresponding to the manipulator, as long as no joint is close to any joint limit. Weighting w_i high ensures that the corresponding elements of $\boldsymbol{\zeta}$ is smaller, thus the velocity of the vehicle is close to zero as long as the manipulator joints are away from its limits.

We will now use a method of controlling the vehicle's motion, inspired from the notion of primary and secondary tasks presented in Antonelli [2013]. Since the system is redundant, one can project an arbitrary velocity vector into the null space of the Jacobian matrix . Informally, this means that, as long as the system has its 12 degrees of freedom (away from joint limits), one can decide the inner motion of the system that does not cause motion of the end effector. This can be done by the following (Liegeois [1977])

$$\boldsymbol{\zeta}_d = \boldsymbol{J}_{e0}^{\dagger}\boldsymbol{V}_{0e} + (\boldsymbol{I} - \boldsymbol{J}_{e0}^{\dagger}\boldsymbol{J}_{e0})\boldsymbol{\zeta}_a \tag{3.43}$$

where $\boldsymbol{\zeta}_a$ is an arbitrary system velocity. The matrix $(\boldsymbol{I}-\boldsymbol{J}_{e0}^{\dagger}\boldsymbol{J}_{e0})$ then projects the velocity $\boldsymbol{\zeta}_a$ into the null space of $\boldsymbol{J}_{e0}$ (Liegeois [1977]). We thus have that

$$(\boldsymbol{I} - \boldsymbol{J}_{e0}^{\dagger}\boldsymbol{J}_{e0})\boldsymbol{\zeta}_a \in Null(\boldsymbol{J}_{e0}), \;\; \forall \boldsymbol{\zeta}_a \in \mathbb{R}^{6+n} \tag{3.44}$$

It then follows that

$$\boldsymbol{J}_{e0}(\boldsymbol{I} - \boldsymbol{J}_{e0}^{\dagger}\boldsymbol{J}_{e0})\boldsymbol{\zeta}_a = \boldsymbol{0} \tag{3.45}$$

and we see that system velocities projected into $Null(\boldsymbol{J}_{e0})$ gives system velocities that causes no motion of the end effector. Since $\boldsymbol{\zeta}_a$ can be any system velocity, we can project the vehicle velocity $\boldsymbol{\nu}$ by using the Jacobian $\boldsymbol{J}_v$

$$\boldsymbol{\nu} = \boldsymbol{J}_v\boldsymbol{\zeta}_v \tag{3.46}$$

$$\boldsymbol{J}_v = \begin{bmatrix} \boldsymbol{I}_{6\times 6} & \boldsymbol{0}_{6\times 6} \end{bmatrix} \tag{3.47}$$

where $\boldsymbol{\zeta}_v$ is the system velocity corresponding to the vehicle velocity $\boldsymbol{\nu}$. It is then easy to see that the relationship $\boldsymbol{\zeta}_v = \boldsymbol{J}_v^{\top}\boldsymbol{\nu}$ holds. One can then write (3.43) as

$$\boldsymbol{\zeta}_d = \boldsymbol{J}_{e0}^{\dagger}\boldsymbol{V}_{0e} + (\boldsymbol{I} - \boldsymbol{J}_{e0}^{\dagger}\boldsymbol{J}_{e0})\boldsymbol{J}_v^{\top}\boldsymbol{\nu}_d \tag{3.48}$$

From this, one can obtain the desired configuration $\boldsymbol{\xi}_d$, which will act as a reference signal to the low level controller, through the following:

$$\boldsymbol{\xi}_d = \int_0^t \boldsymbol{J}_{a,s}\boldsymbol{\zeta}_d \tag{3.49}$$

In simulations of the method presented above, discontinuities in $\boldsymbol{\zeta}_d$ were experienced. The author is not sure if this is a result of the numerical errors in the simulation process, or if it is from the properties of $\boldsymbol{J}_{e0}^{\dagger}$. The problem was however solved by filtering $\boldsymbol{\zeta}_d$ with a 1st order low pass filter with very low time constants:

$$\boldsymbol{\zeta}_d = \frac{1}{s}\left(\boldsymbol{A}_{\zeta_d}^{-1}(\boldsymbol{\zeta}_d - \boldsymbol{\zeta}_d^{'})\right) \tag{3.50}$$

where $\boldsymbol{\zeta}_d^{'}$ is the output of the inverse kinematics in (3.48), and $\boldsymbol{A}_{\zeta_d}$ is a diagonal matrix which yields the time-constants of the filter along its diagonal. It is then important that the time constants of the filter is small. Large time constants will cause too much deviation from $\boldsymbol{\zeta}_d^{'}$ which again will cause a deviation in the end effector trajectory.

3.2.2 Event Based Vehicle Velocity Kinematic Control

To be able to have an operator controlling the motion of the end effector, the motion of the vehicle must be controlled locally. In the previous section a method for avoiding joint limits were proposed, using the weighted least norm method. From (3.48) we have that the first term $\boldsymbol{J}_{e0}^{\dagger}\boldsymbol{V}_{0e}$ yields the vehicle part of $\boldsymbol{\zeta}_d$ close to zero ($\boldsymbol{\nu}_d \approx \boldsymbol{0}$), due to the weighting of $\boldsymbol{W}$, as long as no joint limits are reached. When a joint limit is reached, the corresponding w_i in $\boldsymbol{W}$ is weighted higher than the vehicle part of $\boldsymbol{W}$, yielding velocity to the vehicle part of $\boldsymbol{\zeta}$. This gives a priority to the objective of avoiding joint limits of the manipulator, over the objective of controlling the vehicle motion.

From the second term on the right side of (3.43), namely $(\boldsymbol{I} - \boldsymbol{J}_{e0}^{\dagger}\boldsymbol{J}_{e0})\boldsymbol{J}_v^{\top}\boldsymbol{\nu}_d$, we see that we can project a desired $\boldsymbol{\nu}_d$ to cause motion of the vehicle as long as this is possible in the null space of $\boldsymbol{J}_{e0}$. We will therefore present a method to obtain the vehicle motion $\boldsymbol{\nu}_d$ based on different criteria.

One of those criteria is to keep the manipulator dexterous. This is obtained by both trying to avoid the joint limits, as well as avoiding that the manipulator is fully stretched out, as illustrated in Fig. 3.3. When the manipulator is fully stretched out, certain end effector trajectories can only be obtained by vehicle motion. Tracking is thus more difficult as it relies on the dynamics of the vehicle, which is slower than that of the manipulator. It is therefore preferable that the end effector is operating within a certain subset of the reachable workspace of the end effector relative to the vehicle. We will therefore assign a subset $\mathcal{W}_s$ that is fixed to the vehicle, and which specifies a boundary for the desired

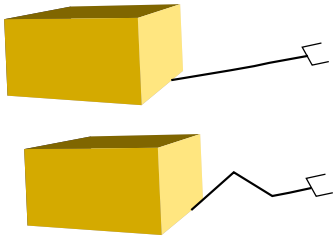


Figure 3.3: The manipulator has lost its dexterity (top) and is fully dexterous (bottom)

position of the end effector relative to the vehicle. We propose that $\mathcal{W}_s$ has the following properties:

- $\mathcal{W}_s$ is visible from cameras attached to the vehicle. This is dependent on the system, but will most likely be in front of the vehicle.
- $\mathcal{W}_s$ does not contain any points where the end effector is fully stretched out or is too close to the vehicle.
- $\mathcal{W}_s$ can be split up into a finite number of convex subsets.

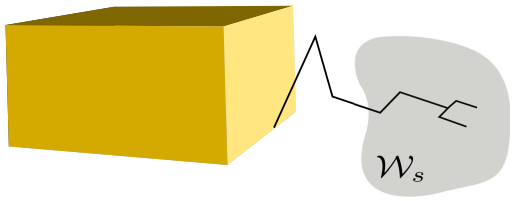


Figure 3.4: Example of the subset $\mathcal{W}_s$. $\mathcal{W}_s$ is fixed relative to the vehicle's frame $\mathcal{F}_b$

One of the objectives of the event based vehicle control system is to control $\boldsymbol{\nu}$ so that the end effector always stays inside $\mathcal{W}_s$. Another objective is to make the end effector point outwards from the vehicle. We therefore construct a set Ψ_s which is the set of allowed angles between the end effector and the vehicle as illustrated in Fig. 3.5.

A third objective is to keep the vehicle in a pose where it uses the least energy. It is both common and reasonable to design the vehicle so that the position of CB (Center of Boyancy) and CG (Center of Mass) makes the vehicle's roll and pitch angles open loop stable around $\phi = \theta = 0^o$. The event based vehicle control system will therefore also return the orientation of the vehicle to $\phi = \theta = 0^o$ when possible.

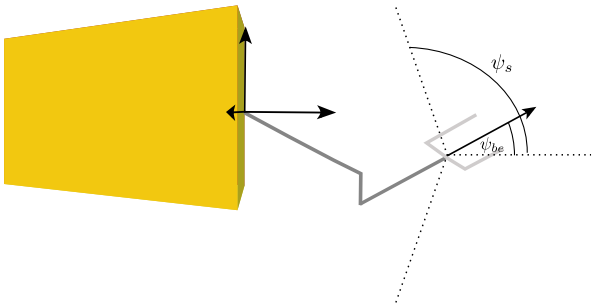


Figure 3.5: Top view of the UVMS illustrating the maximum angle ψ_s of the end effector relative to the vehicle.

To make the system simple and robust, the three problems will be decoupled. This is done by assigning the different DOFs of the vehicle to solve the different problems. We will use the linear velocity part of $\boldsymbol{\nu}$, namely $\boldsymbol{v}_{0b}$ to make end effector stay inside $\mathcal{W}_s$. The angular velocity around the z axis and velocities along the x and y axis of $\mathcal{F}_b$ will be used to make the end effector direct outwards from the vehicle, while the other angular velocities will used to return the vehicle to $\phi = \theta = 0^o$.

3.2.2.1 Staying Inside $\mathcal{W}_s$

We want to make the origin of the end effector frame $\mathcal{F}_e$ stay inside the set of points in $\mathcal{W}_s$. We will let the topology of $\mathcal{W}_s$ be homeomorphic to a sphere, i.e. it must be simply connected, and it must be a closed set. Furthermore we denote the center of $\mathcal{W}_s$ as O_s and attach a frame $\mathcal{F}_s$ to the center, with the same orientation as $\mathcal{F}_b$. We also design a sphere around O_s named $\mathcal{W}_i$, which is completely on the inside of $\mathcal{W}_s$, as illustrated in Fig. 3.6.

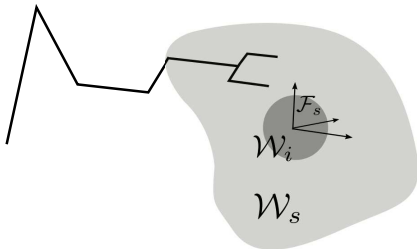


Figure 3.6: Example of $\mathcal{W}_i$ and $\mathcal{W}_s$

One of the important objectives of the control system is to move the vehicle as little as possible. Therefore, when the end effector is inside $\mathcal{W}_s$, there should normally be no linear velocity of the vehicle. A tradeoff to the objective of keeping the vehicle's velocity at zero

is the objective of keeping the end effector in a favorable position relative to the vehicle. Therefore we call $\mathcal{W}_s$ the *acceptable workspace*, and $\mathcal{W}_i$ the *preferred workspace*. If the vehicle is only controlled to stay inside $\mathcal{W}_s$, one can end up with a situation where the end effector always is working close to the boundary of $\mathcal{W}_s$. This is not very optimal, because when working close to the boundary of $\mathcal{W}_s$ the trajectory $\boldsymbol{V}_{e,d}$ is likely to go outside $\mathcal{W}_s$, and one can end up with a situation where the end effector alternates between being inside and outside of $\mathcal{W}_s$, causing excessive control action on the vehicle. It is therefore proposed that when the end effector is outside $\mathcal{W}_s$ it should not only be controlled to reach the inside of $\mathcal{W}_s$, but should also reach the inside of $\mathcal{W}_i$. We now construct a vector $\boldsymbol{p}_{se} \in \mathbb{R}^3$ that describes the position of the end effector in the $\mathcal{F}_s$ frame. We can then define $\mathcal{W}_i$:

$$\mathcal{W}_i = \left\{ \boldsymbol{p}_{se} \in \mathbb{R}^3 \;\middle|\; \|\boldsymbol{p}_{se}\| < r \right\}$$

For some $r > 0$. We also define the commanded vehicle velocity $\boldsymbol{\nu}_{lin,c}$ associated with keeping the end effector inside $\mathcal{W}_i$

$$\boldsymbol{\nu}_{lin,c} := \begin{bmatrix} \boldsymbol{v}_{lin}^\top & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^6 \tag{3.51}$$

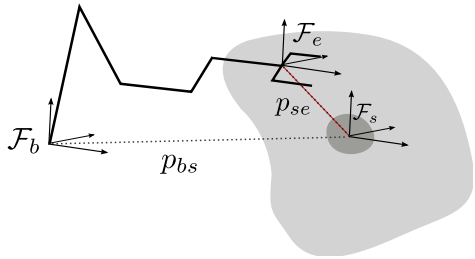


Figure 3.7: Illustration of the vector $\boldsymbol{p}_{se}$ from the origin of $\mathcal{F}_s$ to the origin of $\mathcal{F}_e$

Since $\mathcal{F}_s$ has the same rotation as $\mathcal{F}_b$, one can easily get inside $\mathcal{W}_i$, as long as the vehicle is not rotating, by the following commanded velocity $\boldsymbol{\nu}_{lin,c}$

$$\boldsymbol{\nu}_{lin,c} = \begin{bmatrix} \boldsymbol{R}_{be}\boldsymbol{v}_{0e} + k_{lin}\boldsymbol{p}_{se} \\ \boldsymbol{0} \end{bmatrix} \tag{3.52}$$

where $k_{lin} > 0$ is a tuning parameter that controls how fast the vehicle moves relative to the end effector in the direction of $\boldsymbol{p}_{se}$. Since $\mathcal{F}_s$ is fixed in frame $\mathcal{F}_b$, $\boldsymbol{v}_{0s} = \boldsymbol{v}_{0e}$ as long as the vehicle has no angular velocity. When $\mathcal{F}_b$ is rotating, $\boldsymbol{v}_{0s} = \boldsymbol{v}_{0b} + \widehat{\boldsymbol{p}_{bs}}\boldsymbol{\omega}_{0b}$, where $\boldsymbol{p}_{bs}$ is the constant vector describing the position of $\mathcal{F}_s$ relative to $\mathcal{F}_b$. We can then rewrite (3.52) to include the angular velocity of $\mathcal{F}_b$

$$\boldsymbol{\nu}_{lin,c} = \begin{bmatrix} \boldsymbol{R}_{be}\boldsymbol{v}_{0e} + k_{lin}\boldsymbol{p}_{se} - \widehat{\boldsymbol{p}_{bs}}\boldsymbol{\omega}_{0b} \\ \mathbf{0} \end{bmatrix} \tag{3.53}$$

By using $\boldsymbol{\nu}_{lin,c}$ in (3.53) as the commanded velocity, the origin of $\mathcal{F}_s$ will approach the origin of $\mathcal{F}_e$ with the velocity $k_{lin}\boldsymbol{p}_{se}$, and thus the end effector will reach the inside of $\mathcal{W}_i$. When the end effector reaches the outside of $\mathcal{W}_s$, it is not always optimal to return the end effector to $\mathcal{W}_i$ immediately. Let the end effector travel in one direction over some period of time. Commanding the vehicle to return the end effector to $\mathcal{W}_i$ as soon as it goes outside $\mathcal{W}_s$ would lead to the following behavior: The vehicle will start to move when the end effector reaches the outside $\mathcal{W}_s$, in order to move it back to $\mathcal{W}_i$. Subsequently, the vehicle will stay still when reaching $\mathcal{W}_i$, the end effector will continue, and reach the outside of $\mathcal{W}_s$. Consequently, the vehicle will start moving again. We will therefore experience that the vehicle alternates between being in motion and settling to rest, even though the end effector is constantly moving in one direction. We therefore propose that we let the vehicle move with the same velocity as the end effector (relative to $\mathcal{F}_0$) from the moment the end effector reaches the outside of $\mathcal{W}_s$. Only when the end effector's velocity is zero or has a velocity vector pointing towards O_s (the origin of $\mathcal{F}_s$) the vehicle will be commanded to move the end effector back to $\mathcal{W}_i$. To check if the end effector is moving towards O_s one can project the linear velocity of the end effector (described in $\mathcal{F}_b$), $\boldsymbol{v}_{0e}^b$ along the vector $\boldsymbol{p}_{se}$ using the inner product. If we let $\alpha_v \in \mathbb{R}^1$ denote the inner product

$$\alpha_v = (\boldsymbol{v}_{0e}^b)^\top \boldsymbol{p}_{se} \tag{3.54}$$

one can simply check the sign of α_v to check if the end effector is moving towards O_s. If $\alpha_v > 0$ the end effector has a velocity vector pointing away from O_e.

The proposed algorithm uses a number of if-else statements to check which velocity to assign to the vehicle based on the above theory, and is listed in Algorithm 1.

Algorithm 1 Algorithm for keeping end effector inside $\mathcal{W}_s$

End effector inside $\mathcal{W}_s$
$\nu_{lin,c} \leftarrow 0$
$returnToW_i \leftarrow false$
loop
  $\alpha_v \leftarrow (v_{0e}^b)^\top p_{se}^b$
  if EE is inside $\mathcal{W}_i$ **then**
    $returnToW_i \leftarrow false$
    $\nu_{lin,c} \leftarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
  else if EE is outside $\mathcal{W}_s$ **AND** $\alpha_v >= 0$ **then**
    $\nu_{lin,c} \leftarrow \begin{bmatrix} R_{be} v_{0e} - \widehat{p_{bs}} \omega_{0b} \\ 0 \end{bmatrix}$
    $returnToW_i \leftarrow true$
  else if $returnToW_i = true$ **then**
    $returnToW_i \leftarrow true$
    $\nu_{lin,c} \leftarrow \begin{bmatrix} R_{be} v_{0e} - \widehat{p_{bs}} \omega_{0b} + k_{lin} p_{se} \\ 0 \end{bmatrix}$
  end if
end loop

Checking if the end effector is inside $\mathcal{W}_i$ can easily be done by checking the length of $\boldsymbol{p}_{se}$. Checking if the end effector is inside or outside $\mathcal{W}_s$, however, is not a trivial task, and therefore we present a method to do so in the section below.

3.2.2.2 Checking if the End Effector is Inside $\mathcal{W}_s$

Since $\mathcal{W}_s$ is a subset of the reachable workspace for the end effector, its shape is specified by the specific manipulator. We therefore propose a general method of checking if the end effector is inside $\mathcal{W}_s$. Let $p_i \in \mathbb{R}^3$ denote a point relative to $\mathcal{F}_s$. We then sample a set of points on the surface of $\mathcal{W}_s$ and denote the set of samples P. One can then freely assign points p_i that reflects the boundary of $\mathcal{W}_s$, depending on the kinematics of the specific manipulator. However, it is important that every point $p_i \in \mathcal{W}_s$ is inside the *dexterous* workspace of the manipulator. The dexterous workspace of a manipulator is defined as all points that the end effector can reach with any given orientation. This is illustrated in Fig. 3.8.

Additionally, it is important that the set of samples covers the surface so that no big patches of the surface is without any sampled points.

From the set P one can cover the surface of $\mathcal{W}_s$ in triangles. It is important that the surface mesh is watertight, i.e. that the triangles cover the surface completely. Poisson

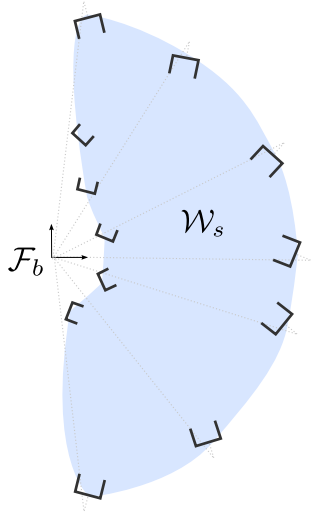


Figure 3.8: Example of sampling of $\mathcal{W}_s$ from the dexterous workspace of the manipulator.

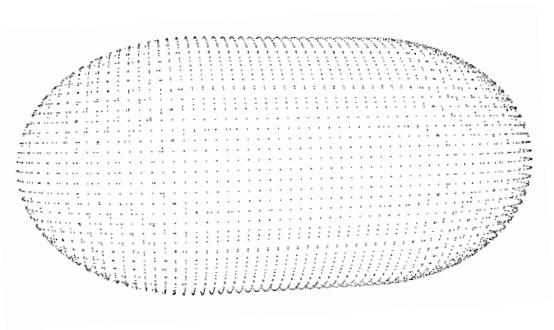


Figure 3.9: Example of sampled points P on the boundary of some $\mathcal{W}_s$.

Surface Construction (Kazhdan et al. [2006]) is able to do just that. Poisson Surface Construction was tested on the point cloud illustrated in Fig. 3.9, and the result is illustrated in Fig. 3.10. It should be noted that the process of generating the mesh is done off-line,

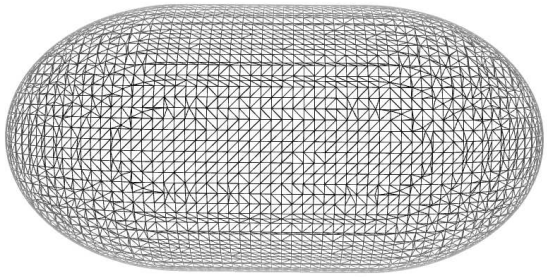


Figure 3.10: A polygon mesh of triangles from the samples in P

and could therefore be done in several ways, as long as the result is a watertight mesh of triangles.

With a set of triangles covering the surface, one can check if the end effector is inside or outside $\mathcal{W}_s$ by casting a ray from the origin of $\mathcal{F}_e$, namely O_e, in an arbitrary direction (see Fig. 3.11) and then count how many triangles the ray intersects. If the ray intersects an odd number of triangles, O_e is inside $\mathcal{W}_s$, and if the ray intersects an even number of triangles it is outside $\mathcal{W}_s$.

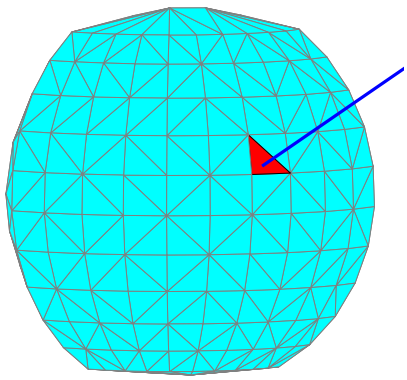


Figure 3.11: Illustration of the ray intersecting one of the triangles (red) of the mesh

Even with a high number of triangles covering the surface, this method is fairly fast. The method was implemented in C++, and was tested on a mesh generated with the Poisson Surface Construction in the Point Cloud Library (Rusu and Cousins [2011]). Timing of the method was done on a desktop PC with and Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz running 32 bits Ubuntu 12.04. The mesh was covered with 14239 triangles. The method of checking if a point was inside or outside the closed mesh took an average of $0.009s$. The implementation of the method was done without any kind of parallel programming. It should be noted, however, that checking intersections between a ray and a set of triangles is very fitting for implementation in a parallel software/hardware setup. One can therefore decrease the speed of the method by increasing the number of processor-cores, if needed.

3.2.2.3 Staying Inside Ψ_s

Although all points inside $\mathcal{W}_s$ are reachable by the end effector, it is not possible to obtain any arbitrary rotation of the end effector inside $\mathcal{W}_s$. Because of limits in the manipulator joints it could, for instance, be difficult to keep the end effector pointed towards the vehicle as illustrated in Fig. 3.12. A method is therefore presented to deal with this problem by

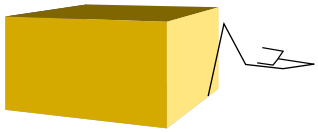


Figure 3.12: Illustration of the end effector pointing towards the vehicle

rotating the vehicle around the origin of $\mathcal{F}_e$. First off, we construct a plane $\mathcal{P}_b$ that is spanned by the x and y -axis of $\mathcal{F}_b$. We then project the center of $\mathcal{F}_e$, namely $\mathcal{O}_e$ onto the plane, together with the axis of $\mathcal{F}_e$ pointing out from the end effector, which in the case of our manipulator is the z_e axis. Furthermore, we construct an angle ψ_{be} which is the angle between the projection of z_e onto $\mathcal{P}_b$ and the axis of $\mathcal{F}_b$ which points outwards (in our case the x_b-axis), see Fig. 3.13. We use the same methodology as in the case of the linear velocity and construct a set of acceptable and preferred angles, Ψ_s and Ψ_i , defined as

$$\Psi_s = \left\{ \psi_{be} \in S^1 \middle| |\psi_{be}| \leq \psi_s \right\} \tag{3.55}$$

$$\Psi_i = \left\{ \psi_{be} \in S^1 \middle| |\psi_{be}| \leq \psi_i \right\} \tag{3.56}$$

$$0 < \psi_i < \psi_s \tag{3.57}$$

Similar to the previous section, we want to keep ψ_{be} inside Ψ_s , and moreover, have ψ_{be} return to Ψ_i when ψ_{be} reaches the boundary of Ψ_s . Let $\boldsymbol{o}_e \in \mathbb{R}^2$ be the projection of $\mathcal{O}_e$ onto $\mathcal{P}_b$ described by the x and y coordinates in $\mathcal{F}_b$. To find the angle ψ_{be} we first project the outward vector of the end effector onto $\mathcal{P}_b$. In our case this is the unit vector along z_e, which we denote $\boldsymbol{r}^e$. We can then represent this vector in $\mathcal{F}_b$ by

$$\boldsymbol{r}^b = \boldsymbol{R}_{be}\boldsymbol{r}^e \tag{3.58}$$

let $\boldsymbol{r}'^b \in \mathbb{R}^2$ be the same vector as $\boldsymbol{r}^b$ only without the $z-$component. $\boldsymbol{r}'^b$ is then a vector lying on $\mathcal{P}_b$ representing the direction of the end effector in $\mathcal{P}_b$, see Fig. 3.14. The angle

ψ_{be} can then be found from

$$\psi_{be} = atan2(y, x) \tag{3.59}$$

where $atan2(y, x)$ is the four-quadrant arctangent of the real parts of x and y satisfying (Fossen [2011]),

$$-\pi \leq atan2(y, x) \leq \pi \tag{3.60}$$

and x and y is the first and second component of $\boldsymbol{r}'^{b}$ respectively.

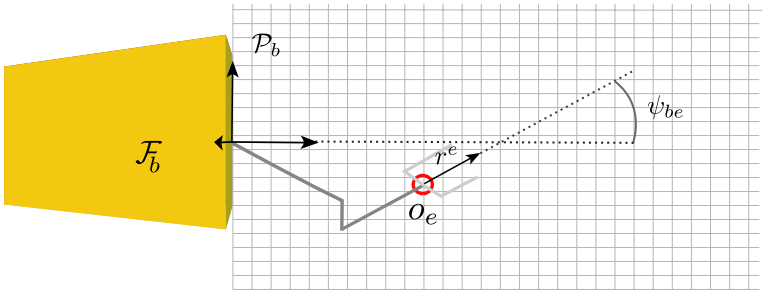


Figure 3.13: Top view of the UVMS, illustrating the angle ψ_{be}

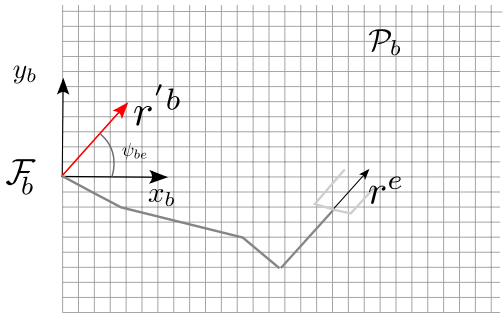


Figure 3.14: Illustration of $\boldsymbol{r}'^{b}$

To keep ψ_{be} inside Ψ_s , we want to move the vehicle around $\boldsymbol{o}_e$. Let r^b_{0b} and r^b_{0e} be the angular velocity of the vehicle and end effector around the z-axis of $\mathcal{F}_b$, defined by:

$$r^b_{0b} := \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \boldsymbol{\omega}^b_{0b}$$
$$r^b_{0e} := \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \boldsymbol{R}_{be} \boldsymbol{\omega}^e_{be}$$

$\dot{\psi}_{be}$ is then the angular velocity of the end effector, relative to the vehicle, projected onto $\mathcal{P}_b$, and thus

$$\dot{\psi}_{be} = r^b_{0e} - r^b_{0b} \tag{3.61}$$

Furthermore, we define the linear velocity $\boldsymbol{v}_{ang} := \begin{bmatrix} u_{ang} & v_{ang} & w_{ang} \end{bmatrix}^\top$ as the linear vehicle velocity corresponding to any r^b_{0e} needed to move in a circle around $\boldsymbol{o}_e$, see Fig. 3.15. The linear velocity $\boldsymbol{v}_{ang}$ corresponding to an angular velocity r^b_{0b} is then

$$\boldsymbol{v}_{ang} = \begin{bmatrix} y_{be} \\ -x_{be} \\ 0 \end{bmatrix} r^b_{0b} \tag{3.62}$$

where x_{be} and y_{be} is the x and y coordinates, respectively, of the point $\boldsymbol{o}_e$. The following velocity then assigns the vehicle velocity which moves the vehicle around $\boldsymbol{o}_e$ and thus keeps $\dot{\psi}_{be} = 0$

$$\boldsymbol{\nu} = \begin{bmatrix} y_{be} \\ -x_{be} \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} r^b_{0e} \tag{3.63}$$

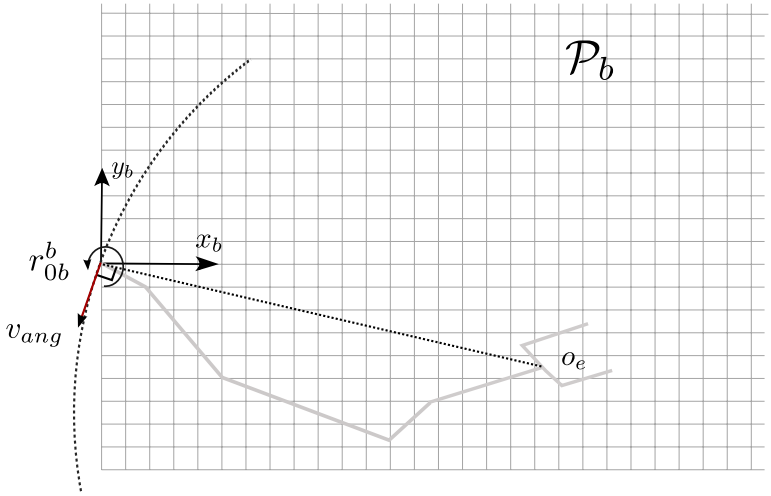


Figure 3.15: Illustration of $\boldsymbol{v}_{ang}$

Let $\boldsymbol{\nu}_{ang,c}$ be the commanded velocity of the vehicle associated with controlling ψ_{be} . ψ_{be} can then be commanded to reach 0, or at least reach the inside of Ψ_i by the commanded velocity

$$\boldsymbol{\nu}_{ang,c} = \begin{bmatrix} y_{be} \\ -x_{be} \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} (r_{0e}^{b} + k_{ang}\psi_{be}) \tag{3.64}$$

where the term $k_{ang}\psi_{be}$ gives the rate of change $\dot{\psi}_{be}$, and $k_{ang} > 0$ is a tuning parameter for how fast ψ_{be} should return to Ψ_i .

The total algorithm for controlling ψ_{be} is listed in algorithm (2):

Algorithm 2 Algorithm for controlling the angle ψ_{be}

ψ_{be} inside ψ_s
$\nu_{ang,c} \leftarrow 0$
returnTo$\Psi_i \leftarrow false$
loop
  if $|\psi_{be}| < \psi_i$ **then**
    returnTo$\Psi_i \leftarrow false$
    $\nu_{ang,c} \leftarrow 0$
  else if $|\psi_{be}| \geq \psi_s$ **OR** returnToΨ_i is true **then**
    returnTo$\Psi_i \leftarrow true$
    $\nu_{ang,c} \leftarrow \begin{bmatrix} y_{be} & -x_{be} & 0 & 0 & 0 & 1 \end{bmatrix}^{\top} (r_{0e}^{b} + k_{ang}\psi_{be})$
  else
    $\nu_{ang,c} \leftarrow 0$
  end if
end loop

3.2.2.4 Keeping $\phi = \theta = 0$

The two methods above only assign the commanded vehicle velocities in the direction of the x, y, z and ψ coordinates of the vehicle, and the commanded roll and pitch angles (ϕ, θ) of the vehicle should stay at zero. When a joint limit is reached, however, the vehicle is assigned velocities through the pseudo inverse $\boldsymbol{J}_{e0}^{\dagger}$ which can cause the vehicle to roll and pitch. We will therefore use a small feedback signal from the roll and pitch angle, in order to get the vehicle back to $\phi = \theta = 0$. We then define the velocity $\boldsymbol{\nu}_{stab,c}$ as the commanded velocity to stabilize the vehicle's roll and pitch:

$$\boldsymbol{\nu}_{stab,c} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -k_\phi \phi \\ -k_\theta \theta \\ 0 \end{bmatrix} \tag{3.65}$$

where $k_\phi, k_\theta > 0$ are small constants that are tuned so that the commanded velocity moves the vehicle to obtain a zero roll and pitch.

3.2.2.5 Smoothing and Saturation of Vehicle Velocity

The event based vehicle kinematic control now gives three velocities $\boldsymbol{\nu}_{lin,c}, \boldsymbol{\nu}_{ang,c}$ and $\boldsymbol{\nu}_{stab,c}$ that should provide a reference trajectory for the vehicle. Let

$$\boldsymbol{\nu}_s := \boldsymbol{\nu}_{lin,c} + \boldsymbol{\nu}_{ang,c} + \boldsymbol{\nu}_{stab,c}$$

The velocity of the vehicle will be limited by the physical properties of the actuators (e.g. the trusters), which are limited in power. We will therefore use *saturating elements* to keep the velocity $\boldsymbol{\nu}_s$ bounded. This is done by defining a signal $\boldsymbol{\nu}_s^{'}$:

$$\boldsymbol{\nu}_s^{'} = sat(\boldsymbol{\nu}_s) \tag{3.66}$$

where $sat(\boldsymbol{x})$ operating on $\boldsymbol{x} \in \mathbb{R}^n$ is itself a vector in $\mathbb{R}^n$ with the ith element defined as:

$$\text{sat}_i(x_i) = \begin{cases} \text{sgn}(x_i) x_{max} & \text{if } |x| \geq x_{i,max} \\ x_i & \text{otherwise} \end{cases}$$

$\boldsymbol{\nu}_s^{'}$ then yields a vehicle velocity that is only piecewise continuous, with discontinuities when the end effector reaches the outside, of $\mathcal{W}_s$ and Ψ_s , when α_v changes sign, and when the end effector returns to $\mathcal{W}_i$ and Ψ_i . We therefore want to filter the output from the algorithm to give a continuous reference $\boldsymbol{\nu}_d$ to the low level control system. This is obtained by using a *reference model* as described in Fossen [2011]. We can then construct a first order low-pass filter that gives us a $\boldsymbol{\nu}_d$ that is continuously differentiable. We then let our reference system be:

$$\dot{\boldsymbol{\nu}}_d = \boldsymbol{A}_\nu \boldsymbol{\nu}_d + \boldsymbol{\nu}_s^{'} \tag{3.67}$$

If we let $\boldsymbol{A}_\nu$ be a diagonal matrix, we can specify the time constants of the filter with the elements on the diagonal of $\boldsymbol{A}_\nu$. Since the reference system is linear, we can easily obtain the signal $\boldsymbol{\nu}_d$ by the following:

$$\boldsymbol{\nu}_d = \frac{1}{s}\Big(\boldsymbol{A}_\nu^{-1}(\boldsymbol{\nu}_d - \boldsymbol{\nu}_s^{'})\Big) \tag{3.68}$$

3.2.3 Discussion of the Kinematic Control

A kinematic control system has now been proposed for a general UVMS. It is designed with a human operator in mind, but the concept should also work for a high level path planner. It uses the Jacobian relationship between the system velocities and the end effector velocity in a weighted least-norm solution for avoiding joint limits of the manipulator. Furthermore, a vehicle velocity is projected into the null space of the Jacobian, and the total vehicle motion can then be assigned as long as the manipulator has at least 6 degrees of freedom. We then use an event based approach to control the vehicle, so that its configuration is such that the manipulator does not lose any DOFs. This is done by designing a set $\mathcal{W}_s$, which contains only points in the workspace, relative to the vehicle, that are reachable by the end effector, and where the manipulator is fully dexterous. There is no guarantee, however, that the manipulator is dexterous at all times due to the following reasons:

- The manipulator can still reach joint limits when the end effector is inside $\mathcal{W}_s$
- Although the vehicle is commanded to move when the end effector reaches the outside of $\mathcal{W}_s$, it cannot guarantee that the end effector returns to the inside of $\mathcal{W}_s$, or stays sufficiently close to the boundary immediately. A transient period might be experienced before the low level controller (SMC) is able to obtain the desired $\boldsymbol{\nu}_d$.

If the desired end effector velocity $\boldsymbol{V}_{e,d}$ is provided by a human operator, this can be solved through having the operator providing low velocities when the end effector is close to the boundary of $\mathcal{W}_s$, or close to joint limits, and thus yielding a reference signal $\boldsymbol{\nu}_s$ that is feasible for the slow dynamics of the vehicle. If, on the other hand, $\boldsymbol{V}_{e,d}$ is changing too fast, close to the boundary and close to joint limits, the sudden change in $\boldsymbol{\nu}_s$ will be filtered through the low pass filter, and thus the vehicle will not follow the given reference.

It should be noted that one needs to filter the output $\boldsymbol{\nu}_s$ to obtain the smooth reference $\boldsymbol{\nu}_d$, which reflects the dynamics of the vehicle. This filtering does not affect the motion of the end effector because $\boldsymbol{\nu}_d$ is part of the inner motion of the system. The filtering of $\boldsymbol{\zeta}_d$

however, affects the end effector motion, and is only for creating a smooth reference signal $\boldsymbol{\xi}_d$.

3.3 Stability of the Total System

Stability of the SMC-UVMS feedback loop was provided in the section above, yielding asymptotic stability of the error $\widetilde{\boldsymbol{\xi}}$. As illustrated in Fig. 3.1 the kinematic control system gives an outer loop with feedback from the measured configuration $\boldsymbol{\xi}$. We will not provide a proof for the stability for the total system, but it can be argued that the system will be stable as long as the input velocity trajectory $\boldsymbol{V}_{e,d}$ is bounded, and continuously differentiable. For a bounded and continuous input $\boldsymbol{V}_{e,d}$, the output $\boldsymbol{\xi}_d$ is also continuously differentiable and bounded. This can be seen from the weighted least norm pseudo inverse $\boldsymbol{J}_{e0}^{\dagger}$ gives a smooth mapping from $\boldsymbol{V}_{e,d}$ to $\boldsymbol{\zeta}_d$, and moreover, the signal $\boldsymbol{\nu}_d$ that is projected into the null space of $\boldsymbol{J}_{e0}$, is also continuously differentiable and bounded. It should be noted that the upper bound on the signal $\boldsymbol{\nu}_d$ is not dependent on the feedback $\boldsymbol{\xi}$, due to the saturating elements on the vehicle velocity.

It should be noted that in a real application of the system, the actuators such as thrusters and motors in the manipulator joints can only supply a limited force $\boldsymbol{\tau}_c$. For further work, it is therefore recommended that this is taken into account when analyzing the stability of the system.

4 | Simulation

4.1 Kinematic Control

In this section the results of the simulation of the pure kinematic control system is presented. Hence, the dynamics is not included in the simulation. All the simulations in this section were done using a generated end effector velocity trajectory, denoted $\boldsymbol{V}_{e,d}$ as input to the system. This trajectory was generated by first constructing waypoints by running the Matlab scripts *makeWaypoints.m* and *makeOrientationWaypoints.m*, where each script generated points in 3D-space representing the desired position and orientation (in euler angle representation) at fixed times. Furthermore, a smooth trajectory $\boldsymbol{P}(t)_d = \begin{bmatrix} (\boldsymbol{p}_d)^\top & (\boldsymbol{\Theta}_d)^\top \end{bmatrix}^\top$ was generated using the script *timeInterpolate.m*. This script constructs smooth interpolation between the waypoints using cubic spline interpolation. The same script uses the Matlab function *fnder* to obtain the derivative $\dot{\boldsymbol{P}}(t)_d$. The desired end effector velocity $\boldsymbol{V}_{e,d}$ can then be obtained by

$$\boldsymbol{V}_{e,d} = \boldsymbol{J}_{a,e}^{-1}\dot{\boldsymbol{P}}_d \tag{4.1}$$

4.1.1 Staying Inside $\mathcal{W}_s$

We now present the output from the simulation where the end effector follows a trajectory $\boldsymbol{V}_{e,d}$. For now, the trajectory $\boldsymbol{V}_{e,d}$ has a constant angular velocity, and thus, the simulation will prove the effectiveness of the control system's ability to assign the desired vehicle

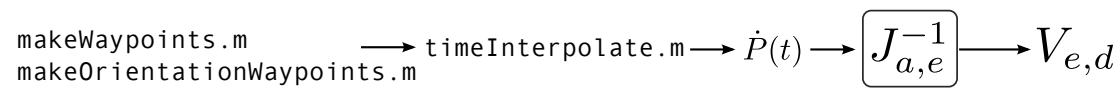


Figure 4.1: Generation of input trajectory $\boldsymbol{V}_{e,d}$

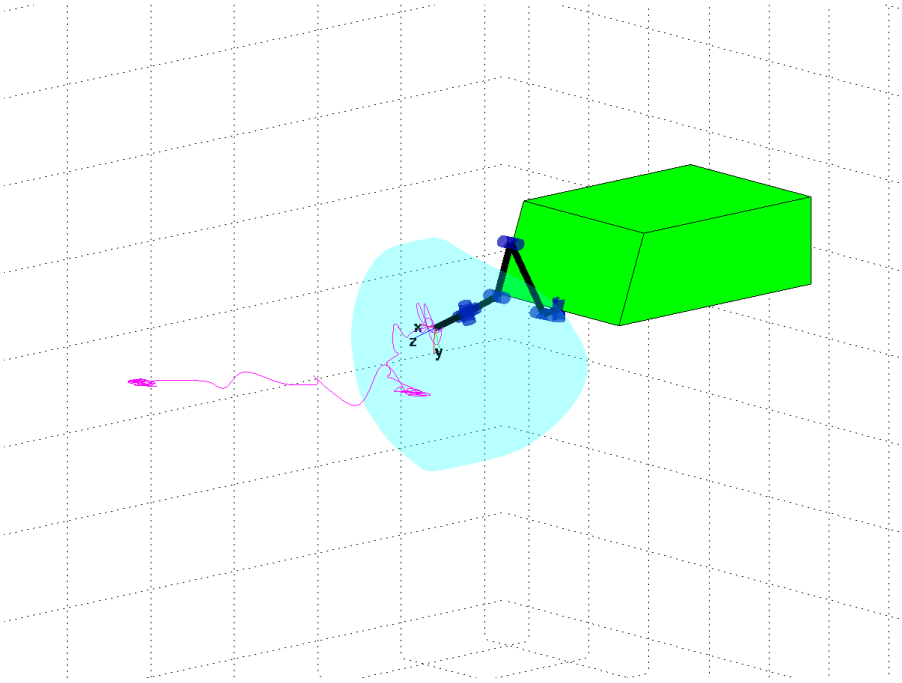
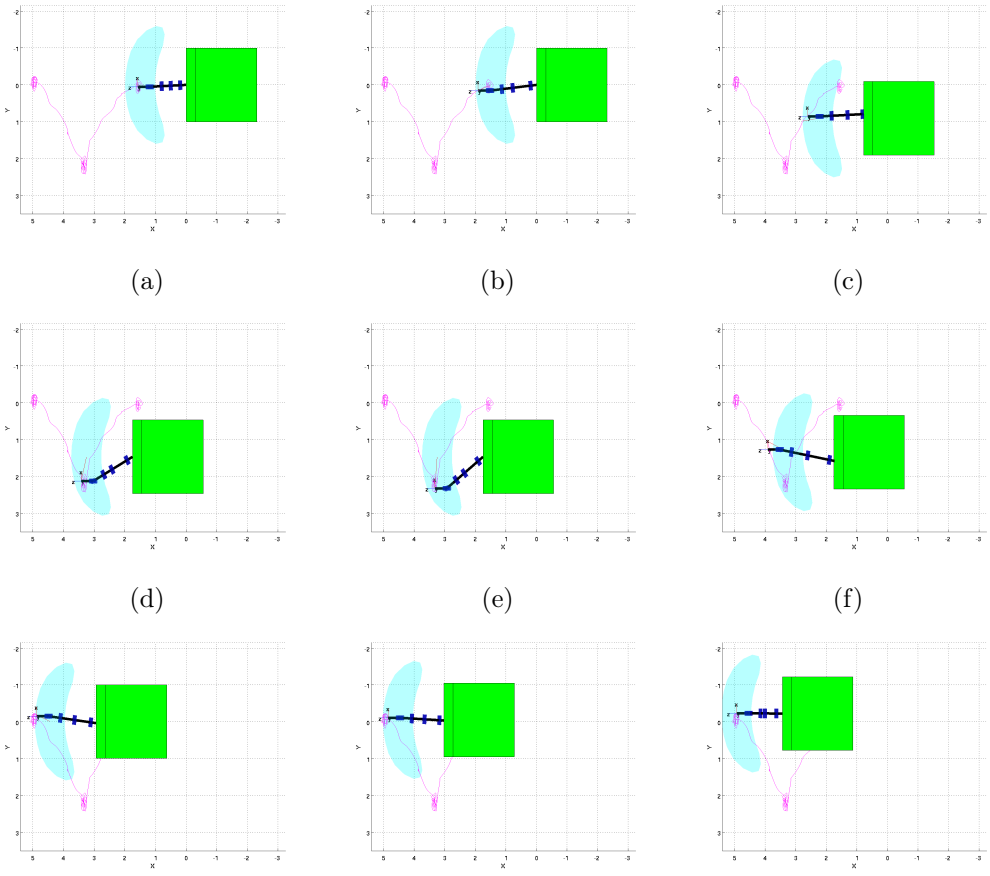


Figure 4.2: Illustration of the trajectory following task with the UVMS in the initial configuration. The end effector is commanded to follow the trajectory in pink. The blue blob illustrates $\mathcal{W}_s$

velocities $\boldsymbol{\nu}_d$ necessary to keep the end effector inside $\mathcal{W}_s$. Fig. 4.2 illustrates the trajectory of the end effector and the UVMS at the initial configuration $\boldsymbol{\xi}(0)$.

In Fig. 4.3 one can observe the evolution of the trajectory tracking. In 4.3a and 4.3b, the vehicle is stationary because the end effector is inside $\mathcal{W}_s$. In 4.3c the end effector has already reached the outside of $\mathcal{W}_s$, and thus, the vehicle moves in the same direction as the end effector. In 4.3d, 4.3e and 4.3f the vehicle is stationary, since the end effector only moves in the inside of $\mathcal{W}_s$. And lastly, in 4.3g the end effector has traveled in the direction of the vehicle leading to a negative α_v in (3.54) (see the previous chapter) and the vehicle moves to get the end effector back to the center of $\mathcal{W}_s$, as seen in 4.3h and 4.3i.

In Fig. 4.4 and Fig. 4.5 one can see the position of the vehicle and end effector over time as the end effector is tracking the trajectory. At approximately $t = 8s$ the end effector reaches the outside of $\mathcal{W}_s$, and thus the vehicle has the same velocity as the end effector. At $t = 12s$ the vehicle is commanded so that the end effector reaches $\mathcal{W}_i$, and at $t = 13s$ the vehicle is stationary. This sequence is repeated from $t = 23s$.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.3: Top view of the uvms tracking the desired path

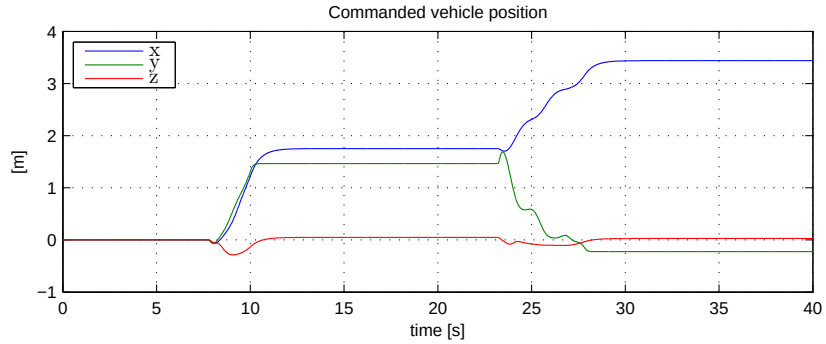


Figure 4.4: Position of vehicle from the event based kinematic vehicle controller

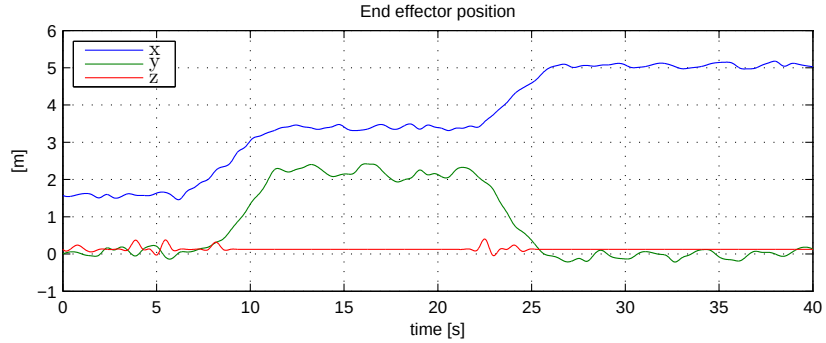


Figure 4.5: Position of the end effector when tracking $\boldsymbol{P}_d$

4.1.2 Staying inside Ψ_s

In this section, the desired end effector trajectory $\boldsymbol{V}_{e,d}$ changes in both linear and angular velocity. As discussed in the previous chapter the vehicle should be commanded to rotate around the center of $\mathcal{W}_s$ if the angle ψ_{be} gets large, i.e. leaves the set Ψ_s . In the subsequent simulation, Ψ_s and Ψ_i is specified as

$$\Psi_s = \{\psi_{be} \in S^1 \Big| |\psi_{be}| < 50^o\} \tag{4.2}$$

$$\Psi_i = \{\psi_{be} \in S^1 \Big| |\psi_{be}| < 10^o\} \tag{4.3}$$

Also, recall that ψ_{be} is the angle between the end effector and the vehicle in the plane spanned by the x and y axes of the vehicle as illustrated in Fig. 4.6.

In Fig. 4.7 one can see the UVMS tracking the desired path $\boldsymbol{V}_{e,d}$, which is changing both in linear and angular velocity. In 4.7a through 4.7c, the vehicle is only translating, and in 4.7d ψ_{be} has left Ψ_s and the vehicle is thus rotating through 4.7e and 4.7f in order

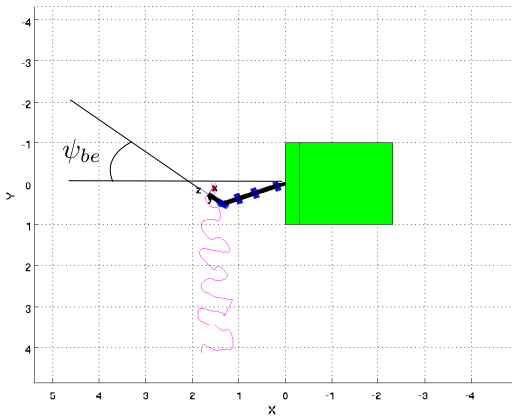
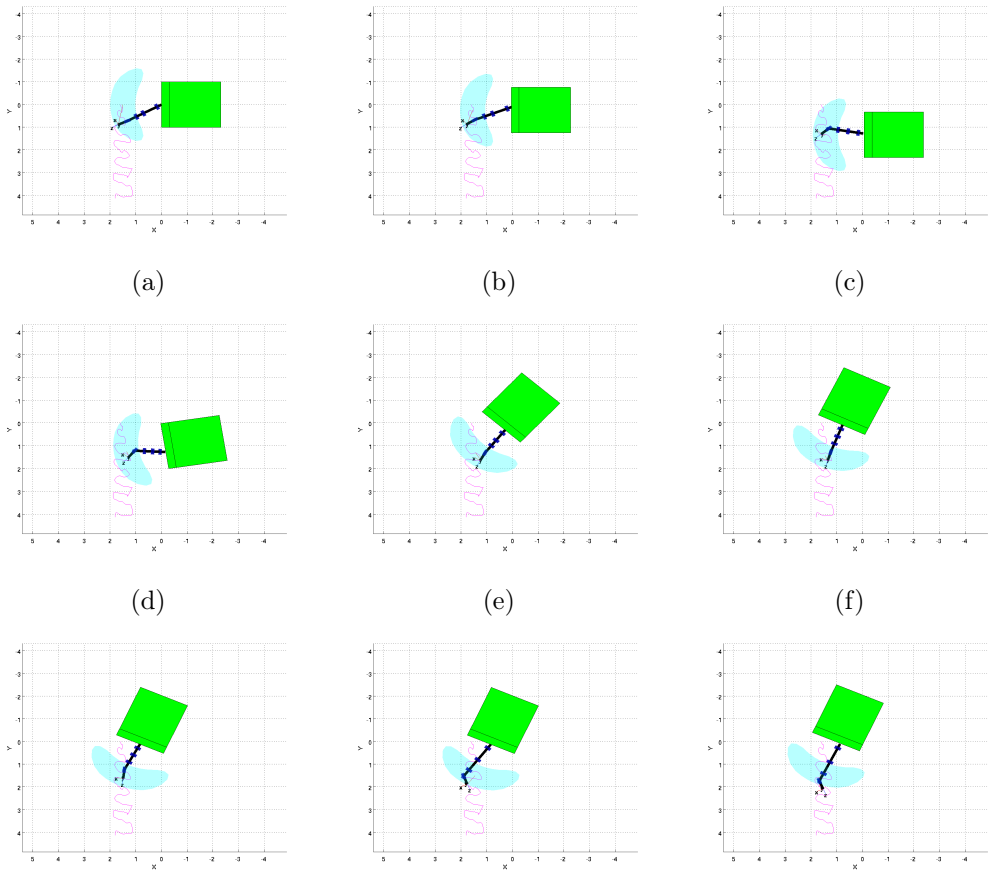


Figure 4.6: Top view of UVMS illustrating ψ_{be}

to bring ψ_{be} back to Ψ_i . Lastly, in 4.7g through 4.7i, the vehicle is again only translating since ψ_{be} is inside Ψ_s .

From the plots below one can see the output of the kinematic control system when responding to changes in orientation of the end effector. The variable ψ_{be} is plotted in Fig. 4.9, and from this one can see that ψ_{be} leaves the set Ψ_s at $t = 11s$, $t = 17s$,$t = 27s$,$t = 33s$ and $t = 37s$. After each time ψ_{be} leaves Ψ_s one can see in Fig. 4.9 that it returns to Ψ_i . This is reflected in Fig. 4.10 with the yaw angle ψ, which is constant as long as ψ_{be} is inside Ψ_s , and changes when ψ_{be} leaves Ψ_s in order to return to Ψ_i . From the plot in Fig. 4.11 one can also see that the x and y coordinates of the vehicle change when ψ changes in order to move in a circle around the end effector.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.7: top view of the UVMS tracking $\boldsymbol{V}_{e,d}$

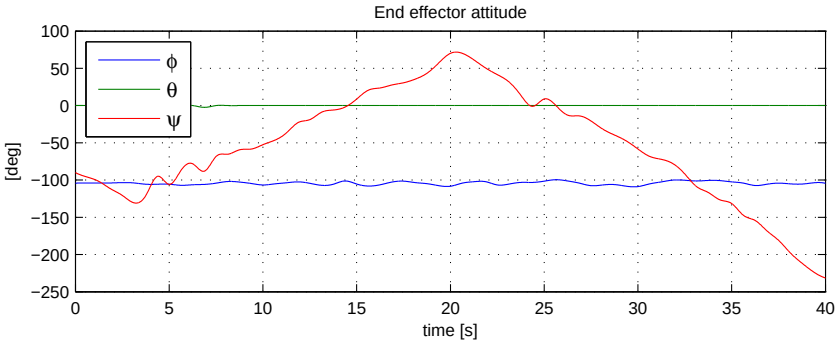


Figure 4.8: Orientation of the end effector ψ_{be}

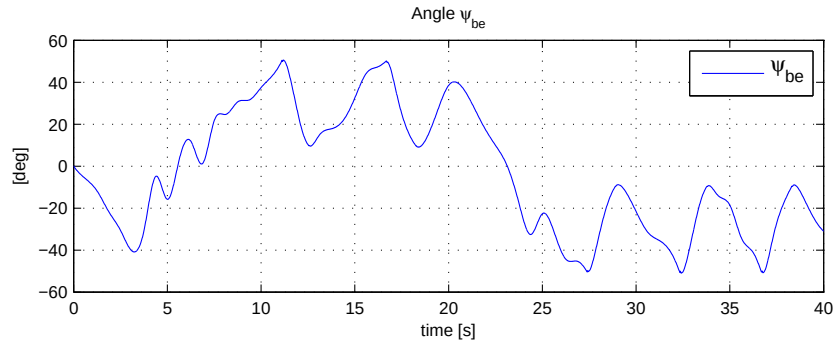


Figure 4.9: Angle between end effector and vehicle ψ_{be}

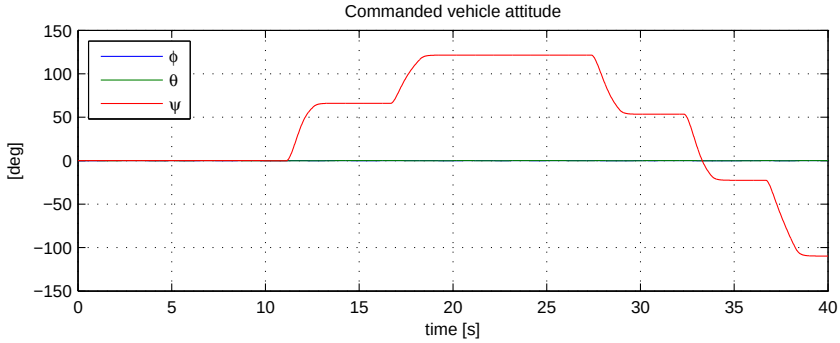


Figure 4.10: Orientation of vehicle

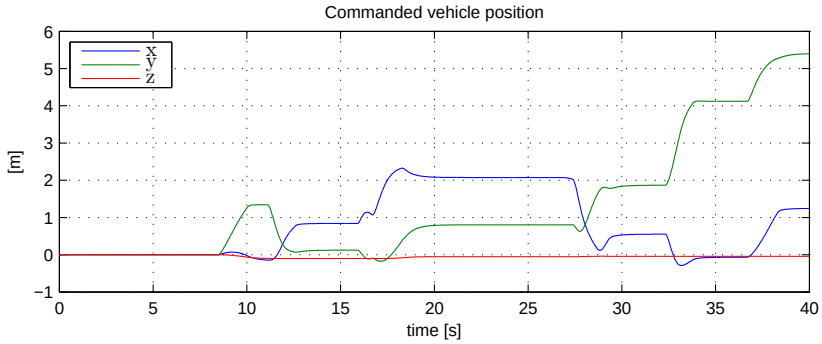


Figure 4.11: Position of vehicle

4.2 Sliding Mode Controller

To show the effectiveness of the sliding mode controller, a simulation is done where the UVMS is tracking smooth reference trajectory directly in the configuration space of $\boldsymbol{\xi}$. Fig. 4.12 illustrates some of the configuration of the UVMS during the simulation. To test the

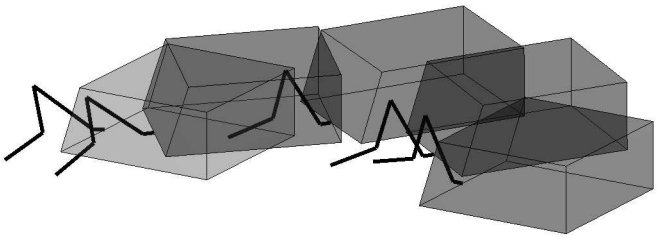


Figure 4.12: Snapshots of the different configuration as the vehicle is both moving in the body $x-axis$ and rotating about the $z-axis$. Two of the links of the manipulator is also rotating while the 4 other are commanded to stay stationary.

robustness of the controller, a current was added to the dynamics of the system, and the dynamics parameters used in the controllers were different from those of the UVMS. The ocean current used in the simulation had the following linear velocity, denoted in $\mathcal{F}_0$:

$$\boldsymbol{v}_c^0 = \begin{bmatrix} 0.4 \\ 0.1 \\ 0 \end{bmatrix} \text{ m/s} \tag{4.4}$$

To simulate a difference between the estimated and real dynamics parameter values the following parameters were used in the controller:

$$\hat{\boldsymbol{M}} = 0.9\boldsymbol{M}$$
$$\hat{\boldsymbol{C}} = 1.2\boldsymbol{C}$$
$$\hat{\boldsymbol{C}} = 1.2\boldsymbol{C}$$
$$\hat{\boldsymbol{D}} = 1.3\boldsymbol{D}$$
$$\hat{\boldsymbol{N}} = 0.75\boldsymbol{N}$$
$$\hat{\boldsymbol{\Xi}} = \mathbf{0}_{12\times 1}$$

where the matrices on the right were used in the simulation of the UVMS dynamics. For the sliding mode controller, the following parameters were used:

$$\lambda = 6$$
$$c = 16$$
$$\boldsymbol{K}_q = diag\{\, 10\ 10\ 10\ 10\ 10\ 10 \,\}$$
$$\boldsymbol{K}_d = diag\{\, 10\ 10\ 10\ 15\ 20\ 15\ 10\ 10\ 10\ 10\ 10\ 10 \,\}$$
$$\boldsymbol{K}_s = diag\{\, 50\ 50\ 50\ 60\ 80\ 60\ 10\ 10\ 10\ 10\ 10\ 10 \,\}$$

The control parameters were simply found through trial and error. In Fig. 4.13 one can see the desired trajectory of the linear vehicle motion together with response of the UVMS to the controller output and ocean current. The controller yields almost perfect tracking, however, a small stationary deviation can be observed in the x-position of the vehicle, which is the direction where the current has the largest velocity component.

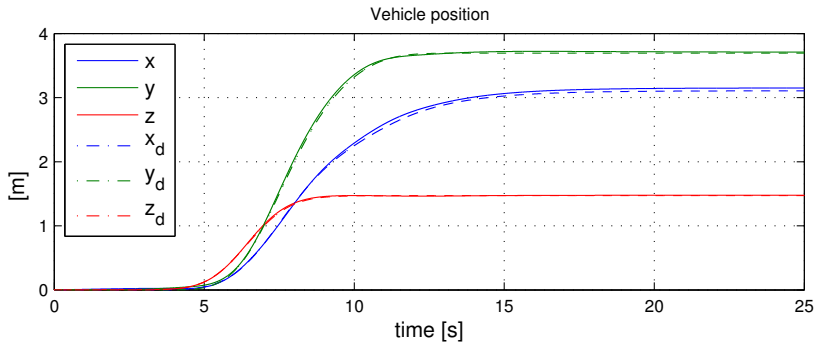


Figure 4.13: The vehicle position is plotted in solid lines and the desired vehicle position is plotted in dotted lines

Fig. 4.14 shows a plot of the desired and measured Euler angles. One can observe almost perfect tracking of the yaw angle ψ. ϕ and θ stays at 0^o with exception of a slight deviation around $t = 5.5s$ where ϕ changes due to coupling forces with the moving manipulator.

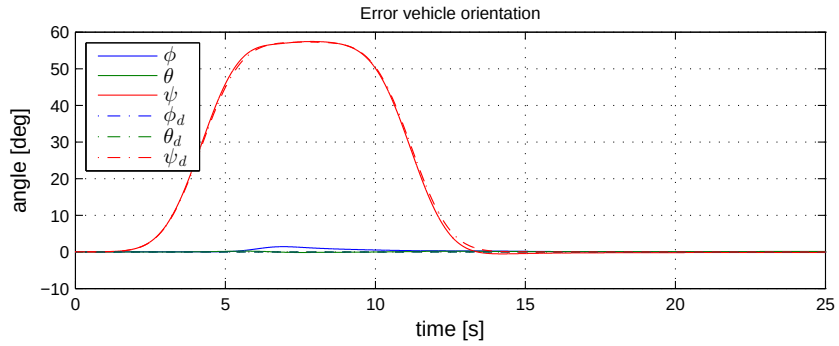


Figure 4.14: The vehicle orientation is plotted in solid lines and the desired vehicle orientation is plotted in dotted lines

For the manipulator, only the two joints q_1 and q_2 are plotted (see Fig. 4.15), as the other joint angles only having a constant desired angle. One can observe good tracking, and an asymptotic behavior where the error gets close to zero when the desired trajectory is constant.

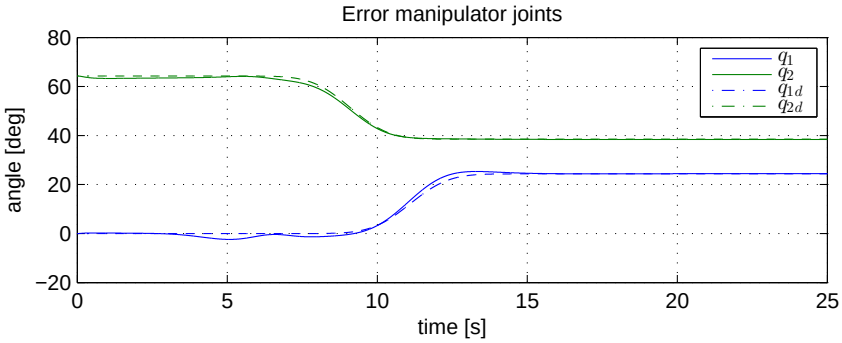
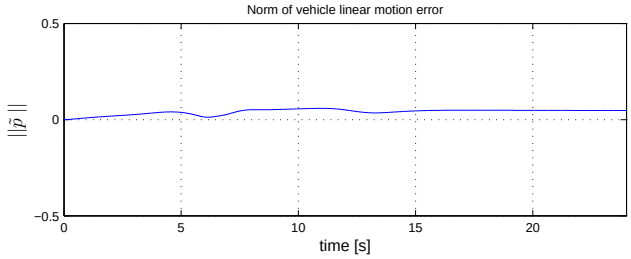


Figure 4.15: The two first manipulator joint angles

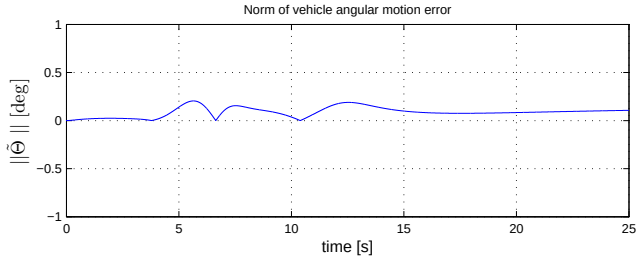
In Fig. 4.16 one can observe the Euclidean norm of the errors for the vehicle linear, vehicle angular, and the manipulator motion. The Euclidean norm of the error vector $\tilde{\boldsymbol{x}}$ at time t is defined as

$$||\tilde{\boldsymbol{x}}(t)|| = \left(\tilde{x}_1(t)^2 + \tilde{x}_2(t)^2 \cdots\right)^{1/2} \tag{4.5}$$

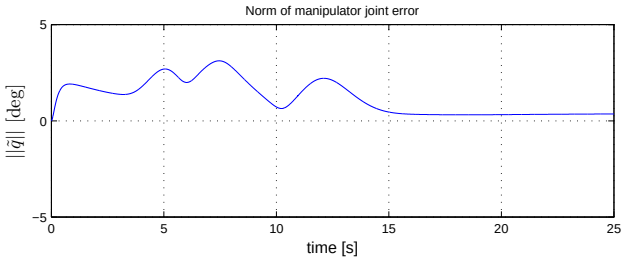
One can observe that the error norm never goes to zero. However, the tracking performance should be satisfactory for most uses, especially considering the influence of the unknown ocean current.



(a) Vehicle linear error



(b) Vehicle angular error



(c) Manipulator error

Figure 4.16: The Euclidean norm of the errors

4.3 Simulation of Kinematic and SMC Control

We will now test the performance of the total control system by testing the performance of both controllers when tracking an end effector trajectory. The control parameters for both controllers are the same as in the simulations above. In Fig. 4.17 and Fig. 4.18 one can observe the end effector and vehicle position. The vehicle is stationary most of the time, but is moving when the end effector reaches the outside of $\mathcal{W}_s$, or when ψ_{be} is reaching the outside of ψ_s , causing the vehicle to move in the x and y direction while also rotation about the z axis. In Fig. 4.21 one can see the norm of the position error of the end effector, which stays within a bound of 10 cm.

In Fig. 4.19 and Fig. 4.20 one can see the end effector and vehicle orientation. The control system yields good tracking of the end effector orientation trajectory, with a norm on the Euler angle error below 2 degrees, see Fig. 4.21.

From Fig. 4.23 one can observe the commanded forces and torques $\boldsymbol{\tau}_c$. In a real application, it is not very realistic that any actuators of the system can obtain such high values, and in the subsequent chapters we will therefore simulate the system where $\boldsymbol{\tau}_c$ is limited to an upper and lower bound, and we will simulate the system with with a low-pass filter on the output of the kinematic control system.

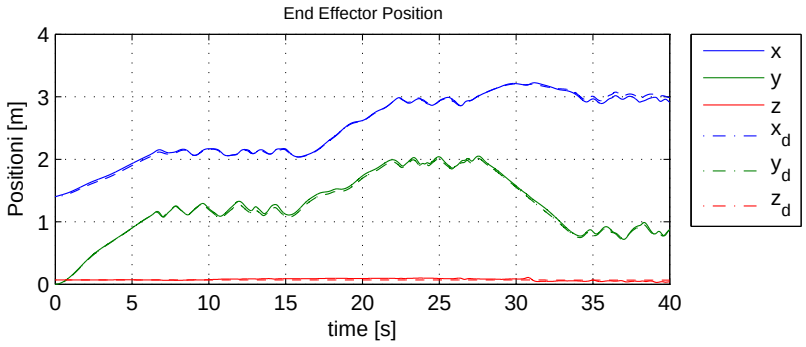


Figure 4.17: End effector measured and commanded position

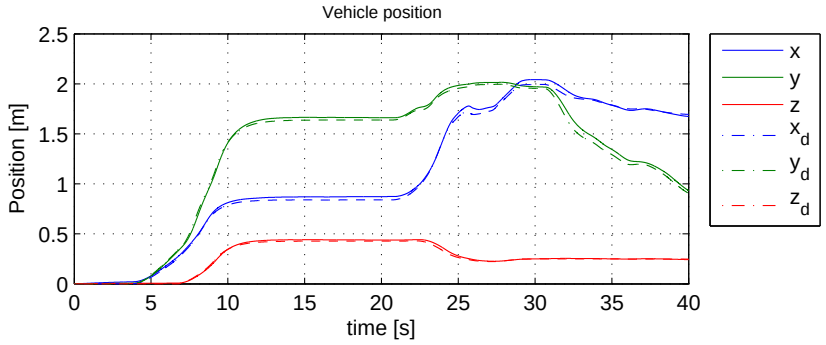


Figure 4.18: Vehicle measured and commanded position

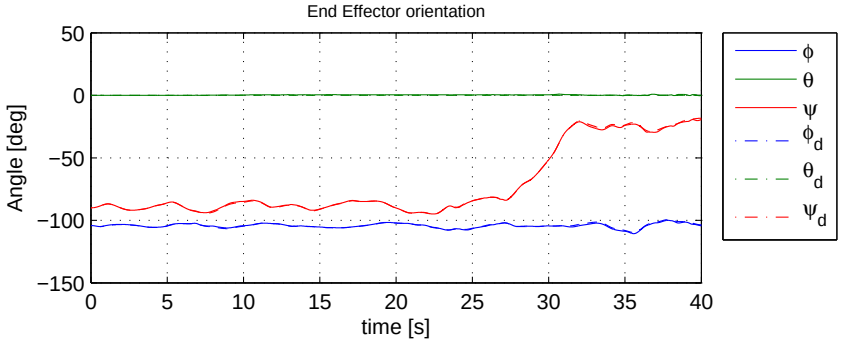


Figure 4.19: End effector measured and commanded orientation

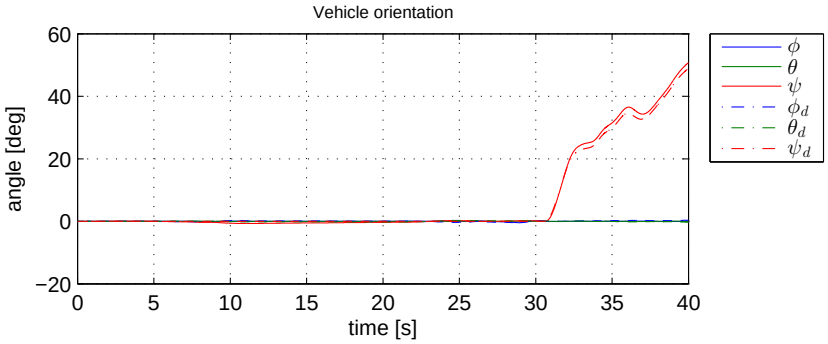


Figure 4.20: Vehicle measured and commanded orientation

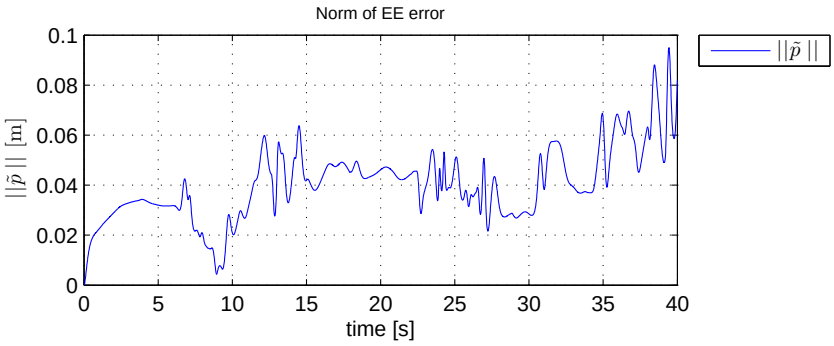


Figure 4.21: Euclidean norm of the end effector position error

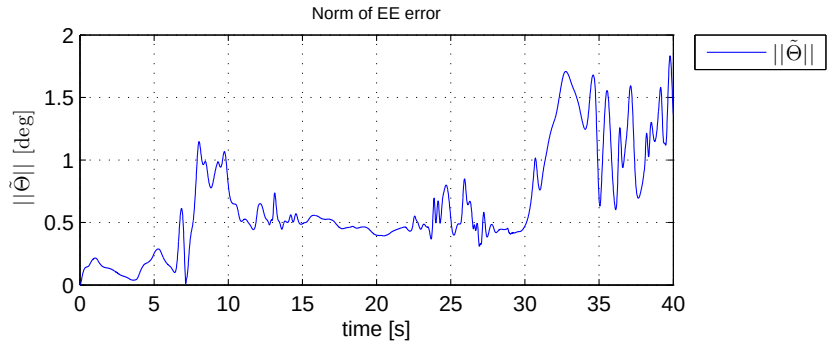


Figure 4.22: Euclidean norm of the end effector orientation error

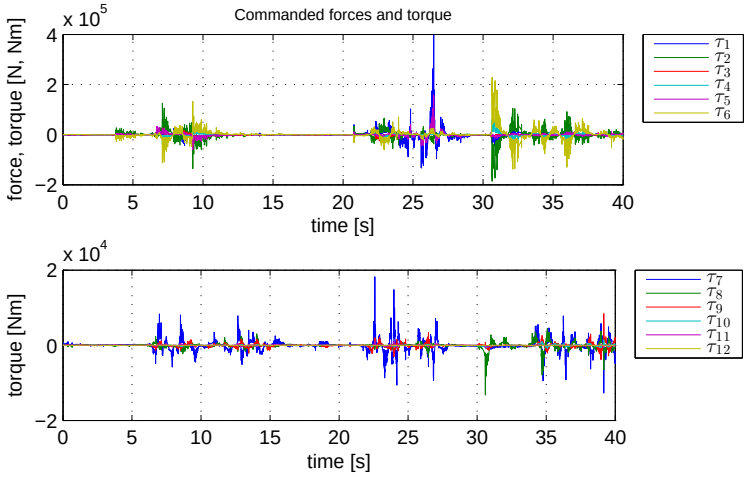


Figure 4.23: Commanded forces and torque $\boldsymbol{\tau}_c$

4.3.1 Simulation of Total System with Limited $\boldsymbol{\tau}_c$

From the previous section we could see that the output from the sliding mode controller, namely $\boldsymbol{\tau}_c$, yielded very high values. The system is therefore simulated with saturation on $\boldsymbol{\tau}_c$ to illustrate a more realistic example. The trajectory and control parameters are the same as in the previous section. We then saturated the torques and forces to be below 4000 N, i.e.:

$$|\tau_{c,i}| \leq 4000\text{N}$$

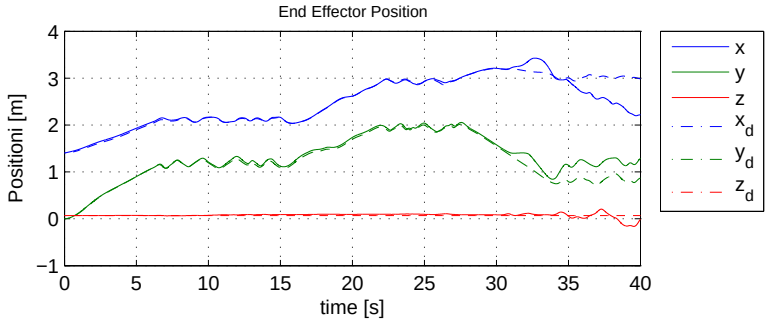


Figure 4.24: End effector position with saturation on $\boldsymbol{\tau}_c$

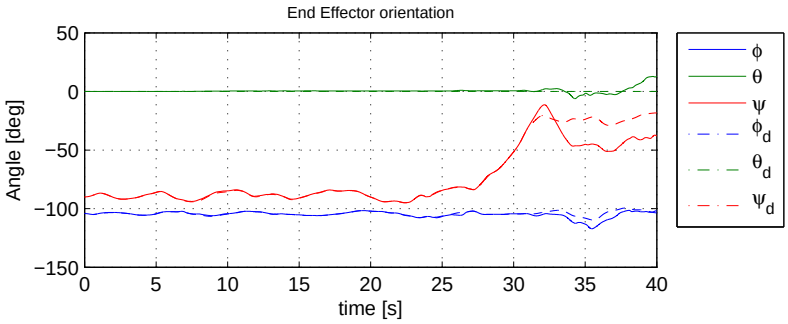


Figure 4.25: End effector orientation with saturation on $\boldsymbol{\tau}_c$

From Fig. 4.24 and Fig. 4.25 we can see that the UVMS is tracking the desired trajectory, while from $t = 32s$ we can see a significant deviation. From Fig. 4.26 we can see that the commanded torque $\boldsymbol{\tau}_c$ is saturated for much of the time, and is changing rapidly, especially at the end.

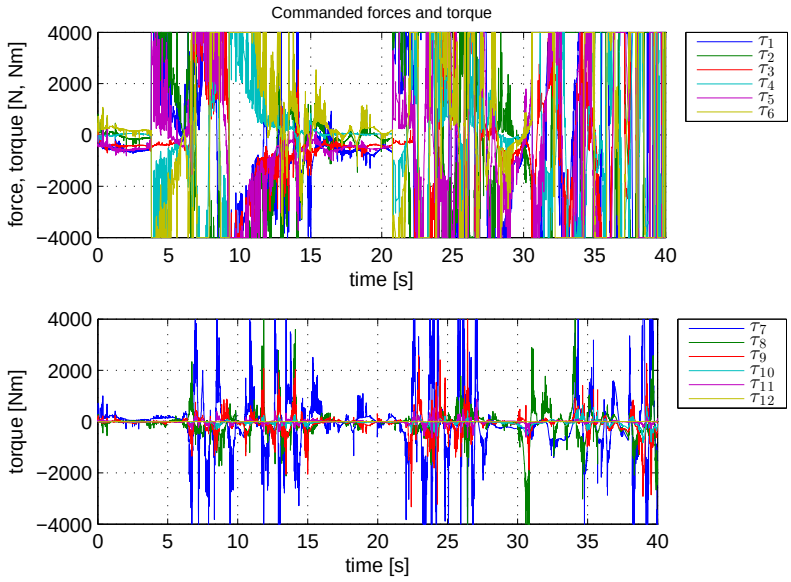


Figure 4.26: Commanded forces and torque from the controller with saturation on $\boldsymbol{\tau}_c$

4.3.2 Simulation of Total System with Filtered $\boldsymbol{\zeta}_d$

From the previous section, we see that the commanded forces and torque $\boldsymbol{\tau}_c$ is changing too rapidly, and yields very high values. The main reason for this is that the output from the kinematic control system, namely $\boldsymbol{\xi}_d$, is not sufficiently smooth, as discussed in Section 3.2. We have therefore simulated the system with a simple 1st order low pass filter applied to $\boldsymbol{\zeta}_d$, thus ensuring sufficient smoothness of $\boldsymbol{\xi}_d$. The following matrix was used for the reference system:

$$\boldsymbol{A}_{\zeta_d} = diag\{0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01\} \tag{4.6}$$

From Fig. 4.27 and Fig. 4.28 one can see that the end effector tracking is still very good. Moreover, Fig. 4.29 shows a continuous $\boldsymbol{\tau}_c$ yielding much lower values than observed in Fig. 4.23. However, the vehicle part of $\boldsymbol{\tau}_c$ still yields values that might be too big to obtain with any thrusters. One can see a big spike in $\tau_{c,1}$ at time $26s$. This is because the manipulator reaches a joint limit, and thus the corresponding part of the weighting matrix $\boldsymbol{W}$ reaches a high value, as can bee observed in Fig. 4.30. The result is that the vehicle is assigned velocities that are meant for the fast dynamics of the end effector. Consequently, the control system gives a high output in order to follow the quick change of $\boldsymbol{\nu}_d$. One solution could be to reject high velocities of the end effector when a joint limit is reached.

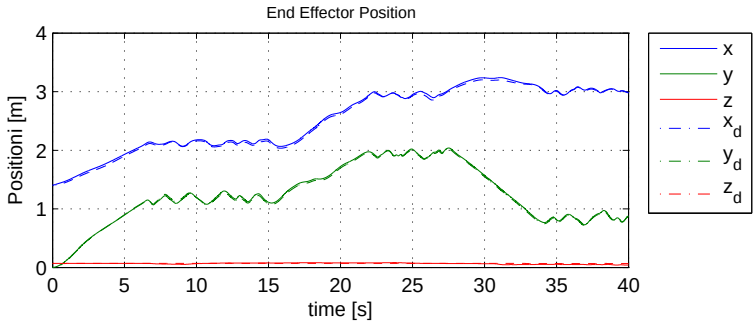


Figure 4.27: Position of the end effector

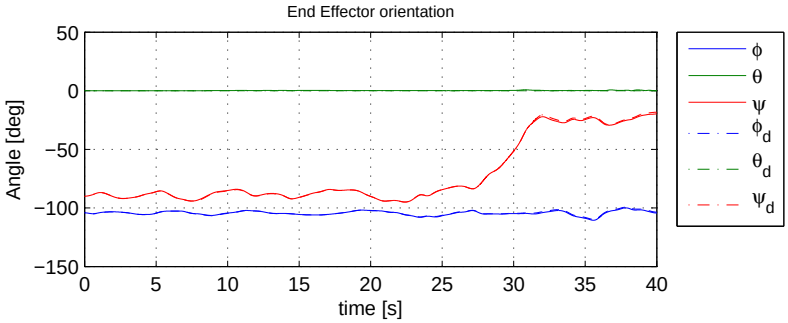


Figure 4.28: Orientation of the end effector

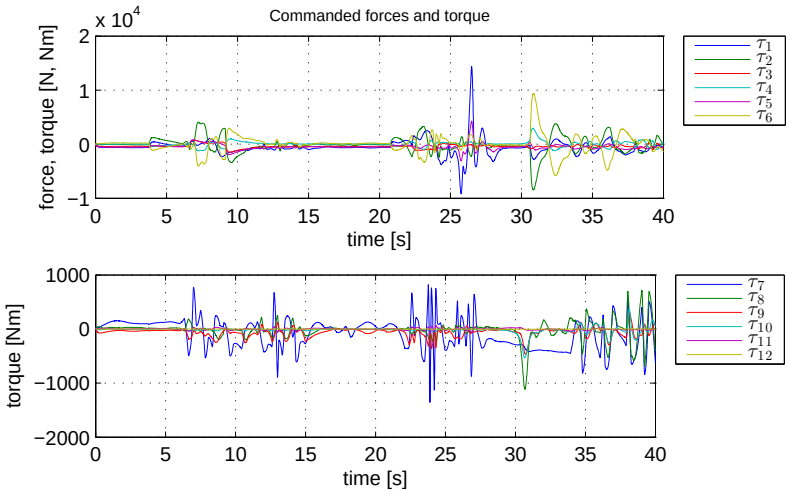


Figure 4.29: Commanded forces and torques

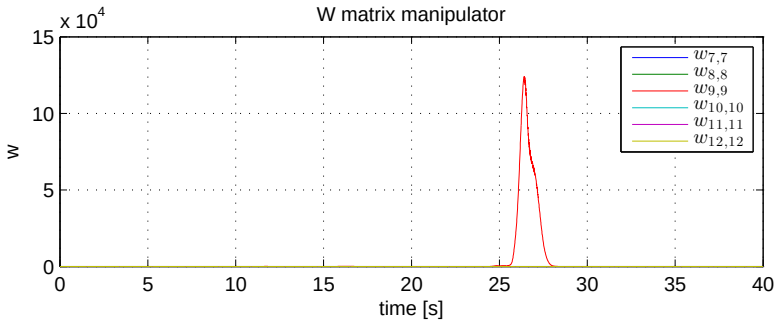


Figure 4.30: Manipulator Part of Weighting Matrix $\boldsymbol{W}$

5 | Conclusion and Further Work

A dynamical model for the UVMS was derived in chapter 2. This includes the dynamics equation (2.126) which yields a compact, closed form description of the dynamics of a UVMS operating underwater with the influence of an ocean current. It was also proved that the different terms in the dynamics equation have some nice properties that can be utilized in control systems design.

Based on the dynamics equation, a sliding mode control law was obtained, which proves to be robust when it comes to uncertainties in both the dynamics parameters and the ocean current, as long as a bounds on the uncertainties exist. Obtaining the bounds on the uncertainties is, however, not specified in this paper, and for further work, it is thus recomdended to find a good method of finding these bounds. This could be based on knowledge of the maximum velocity of the ocean current, maximum allowed system velocities and bounds on the dynamics parameters.

A kinematic control law was designed for facilitating operation of the UVMS system. This was done by using the weighted least-norm pseudo inverse of the geometric Jacobian, which allows an operator to only specify the end effector motion. The weighted least-norm solution also avoids that the manipulator reaches the mechanical joint limits. It is desirable that the pseudo inverse Jacobian maps a continuous end effector velocities to continuous system velocities. In simulations, however, the output was discontinuous. For further work, it is therefore proposed to make sure that the output is continuous.

Furthermore, an event based algorithm were used to decide the motion of the vehicle, based on the position of the end effector relative to the vehicle. This was simulated for arbitrary end effector trajectories, showing good results, were the vehicle was stationary as long as the end effector could follow its trajectory. To decide if the end effector is able to follow the desired trajectory without any vehicle motion, a meshed 3D polygon is attached to the vehicle, and the systems then checks whether the end effector is inside the polygon. The meshed 3D polygon should then specify the workspace of the manipulator, relative to

the vehicle, were the manipulator is dexterous. The polygon is generated by some sort of sampling process, followed by a triangulation of the sample points. For further work, this process should be specified for a general manipulator.

Appendices

A | Derivation of Dynamics Equation

A.1 Differentiation of Jacobian Matrix

The time derivative of the jacobian $\dot{\boldsymbol{J}}_{i0}$ used in the Coriolis matrix

$$\dot{\boldsymbol{J}}_{i0} = \frac{d}{dt}\left(\begin{bmatrix}\boldsymbol{Ad}_{g_{bi}}^{-1} & \boldsymbol{Ad}_{g_{bi}}^{-1}\boldsymbol{J}_i\end{bmatrix}\right) \tag{A.1}$$

$$= \begin{bmatrix}\frac{d}{dt}\boldsymbol{Ad}_{g_{bi}}^{-1} & \frac{d}{dt}\left(\boldsymbol{Ad}_{g_{bi}}^{-1}\right)\boldsymbol{J}_i + \boldsymbol{Ad}_{g_{bi}}^{-1}\frac{d}{dt}\left(\boldsymbol{J}_i\right)\end{bmatrix} \tag{A.2}$$

where

$$\frac{d}{dt}\boldsymbol{Ad}_{g_{bi}}^{-1} = \frac{d}{dt}\begin{bmatrix}\boldsymbol{R}_{bi}^T & -\boldsymbol{R}_{bi}^T\widehat{\boldsymbol{p}}_{bi} \\ 0 & \boldsymbol{R}_{bi}^T\end{bmatrix} \tag{A.3}$$

$$\frac{d}{dt}\boldsymbol{Ad}_{g_{bi}}^{-1} = \begin{bmatrix}\boldsymbol{R}_{bi}^T\widehat{\boldsymbol{\omega}}_{ib}^b & -\boldsymbol{R}_{bi}^T\widehat{\boldsymbol{\omega}}_{ib}^b\widehat{\boldsymbol{p}}_{bi} + \boldsymbol{R}_{bi}^T\dot{\widehat{\boldsymbol{p}}}_{bi} \\ 0 & \boldsymbol{R}_{bi}^T\widehat{\boldsymbol{\omega}}_{ib}^b\end{bmatrix} \tag{A.4}$$

The kth column of $\dfrac{d}{dt}\boldsymbol{J}_i$

$$\left\{\frac{d}{dt}\boldsymbol{J}_i\right\}_k = \begin{bmatrix} * & \dot{\widehat{\boldsymbol{p}}}_{bk}\boldsymbol{R}_{bk} + \widehat{\boldsymbol{p}}_{bk}\boldsymbol{R}_{bk}\widehat{\boldsymbol{\omega}}_{bk}^k \\ * & \boldsymbol{R}_{bk}\widehat{\boldsymbol{\omega}}_{bk}^k\end{bmatrix}\begin{bmatrix}0\\0\\0\\0\\0\\1\end{bmatrix}, \; k \leq i \tag{A.5}$$

$$= \mathbf{0}_{6\times 1}, \; k > i \tag{A.6}$$

Where we have used the properties (see e.g. Fossen [2011])

$$\dot{\boldsymbol{R}}_{ab} = \boldsymbol{R}_{ab}\widehat{\boldsymbol{\omega}}_{ab}^b \tag{A.7}$$

$$\boldsymbol{R}_{bi} = \left(\boldsymbol{R}_{ib}\right)^T \tag{A.8}$$

and where

$$\boldsymbol{\omega}_{bk}^k = \boldsymbol{\omega}_{0k}^k - \left(\boldsymbol{R}_{bk}\right)^T\boldsymbol{\omega}_{0b}^b \tag{A.9}$$

Furthermore we get that

$$\boldsymbol{\omega}_{ib}^{b} = \boldsymbol{\omega}_{0b}^{b} - \boldsymbol{R}_{bi}\begin{bmatrix}\mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3}\end{bmatrix}(\boldsymbol{J}_{gi}^{B}\boldsymbol{\zeta}) \tag{A.10}$$

Proof. This can be seen if we write the expression

$$\begin{aligned}
\boldsymbol{\omega}_{ib}^{b} &= -\boldsymbol{\omega}_{bi}^{b} && \text{(A.11)}\\
&= -\left(\boldsymbol{\omega}_{0i}^{b} - \boldsymbol{\omega}_{0b}^{b}\right) && \text{(A.12)}\\
&= -\boldsymbol{\omega}_{0i}^{b} + \boldsymbol{\omega}_{0b}^{b} && \text{(A.13)}\\
\boldsymbol{\omega}_{0i}^{b} &= \boldsymbol{R}_{bi}\boldsymbol{\omega}_{0i}^{B} && \text{(A.14)}\\
\boldsymbol{\omega}_{0i}^{B} &= \begin{bmatrix}\mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3}\end{bmatrix}\boldsymbol{J}_{gi}^{B}\boldsymbol{\zeta} && \text{(A.15)}
\end{aligned}$$

□

And the time derivative of $\boldsymbol{p}_{bi}$ yields

$$\begin{aligned}
\dot{\boldsymbol{p}}_{bi} &= \boldsymbol{R}_{bi}\boldsymbol{H}\boldsymbol{Ad}_{g_{bi}}^{-1}\boldsymbol{J}_{i}\dot{\boldsymbol{q}} && \text{(A.16)}\\
&\text{where} && \text{(A.17)}\\
\boldsymbol{H} &= \begin{bmatrix}\boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3}\end{bmatrix} && \text{(A.18)}
\end{aligned}$$

Proof. This is obtained by

$$\begin{aligned}
\dot{\boldsymbol{p}}_{bi} &= \boldsymbol{v}_{bi}^{b} && \text{(A.19)}\\
&= \boldsymbol{R}_{bi}\boldsymbol{v}_{bi}^{i} && \text{(A.20)}\\
\boldsymbol{V}_{bi}^{B} &= \boldsymbol{Ad}_{g_{bi}}^{-1}\boldsymbol{J}_{i}\dot{\boldsymbol{q}} && \text{(A.21)}\\
\boldsymbol{v}_{bi}^{B} &= \boldsymbol{H}\boldsymbol{V}_{bi}^{B} && \text{(A.22)}\\
\boldsymbol{v}_{bi}^{b} &= \boldsymbol{R}_{bi}\boldsymbol{v}_{bi}^{B} && \text{(A.23)}
\end{aligned}$$

□

A.2 Derivation of $\boldsymbol{W}_i$

The matrix $\boldsymbol{W}_i$ used in the Coriolis matrix is written below

$$\boldsymbol{W}_i(\boldsymbol{V}_{0i}^B) = \begin{bmatrix} 0 & \widehat{\frac{\partial \mathcal{K}_i}{\partial \boldsymbol{v}_{0i}^B}} \\ \widehat{\frac{\partial \mathcal{K}_i}{\partial \boldsymbol{v}_{0i}^B}} & \widehat{\frac{\partial \mathcal{K}_i}{\partial \boldsymbol{\omega}_{0i}^B}} \end{bmatrix} \tag{A.24}$$

$$\mathcal{K}_i = \frac{1}{2}(\boldsymbol{V}_{0i}^B)^T \boldsymbol{I}_i \boldsymbol{V}_{0i}^B \tag{A.25}$$

$$= \frac{1}{2}\begin{bmatrix}(\boldsymbol{v}_{0i}^B)^T & (\boldsymbol{\omega}_{0i}^B)^T\end{bmatrix}\begin{bmatrix} m\boldsymbol{I}_{3\times 3} & -m\widehat{\boldsymbol{r}}_g^b \\ m\widehat{\boldsymbol{r}}_g^b & \boldsymbol{I}_b \end{bmatrix}\begin{bmatrix}\boldsymbol{v}_{0i} \\ \boldsymbol{\omega}_{0i}^B\end{bmatrix} \tag{A.26}$$

$$= \frac{1}{2}\left(m(\boldsymbol{v}_{0i}^B)^T \boldsymbol{I}_{3\times 3}\boldsymbol{v}_{0i}^B + m(\boldsymbol{\omega}_{0i}^B)^T \widehat{\boldsymbol{r}}_g^b \boldsymbol{v}_{0i}^B - m(\boldsymbol{v}_{0i}^B)^T \widehat{\boldsymbol{r}}_g^b \boldsymbol{\omega}_{0i}^B + (\boldsymbol{\omega}_{0i}^B)^T \boldsymbol{I}_b \boldsymbol{\omega}_{0i}^B\right) \tag{A.27}$$

$$= \frac{1}{2}\left(m(\boldsymbol{v}_{0i}^B)^T \boldsymbol{I}_{3\times 3}\boldsymbol{v}_{0i}^B - 2m(\boldsymbol{v}_{0i}^B)^T \widehat{\boldsymbol{r}}_g^b \boldsymbol{\omega}_{0i}^B + (\boldsymbol{\omega}_{0i}^B)^T \boldsymbol{I}_b \boldsymbol{\omega}_{0i}^B\right) \tag{A.28}$$

$$= \frac{1}{2}\left(m(\boldsymbol{v}_{0i}^B)^T \boldsymbol{I}_{3\times 3}\boldsymbol{v}_{0i}^B + 2m(\boldsymbol{\omega}_{0i}^B)^T \widehat{\boldsymbol{r}}_g^b \boldsymbol{v}_{0i}^B + (\boldsymbol{\omega}_{0i}^B)^T \boldsymbol{I}_b \boldsymbol{\omega}_{0i}^B\right) \tag{A.29}$$

$$\frac{\partial \mathcal{K}_i}{\partial \boldsymbol{v}_{0i}^B} = m\boldsymbol{I}_{3\times 3}\boldsymbol{v}_{0i}^B - m\widehat{\boldsymbol{r}}_g^b \boldsymbol{\omega}_{0i}^B \tag{A.30}$$

$$\frac{\partial \mathcal{K}_i}{\partial \boldsymbol{\omega}_{0i}^B} = \boldsymbol{I}_b \boldsymbol{\omega}_{0i}^B + m\widehat{\boldsymbol{r}}_g^b \boldsymbol{v}_{0i}^B \tag{A.31}$$

Where $\boldsymbol{I}_b$ is the inertia matrix, and $\boldsymbol{r}_g^b$ is the vector from the origin of $\mathcal{F}_i$ to CG of link i. To obtain the partial derivatives we have used the properties

$$\frac{d}{d\boldsymbol{a}}\boldsymbol{c}^T\boldsymbol{a} = \boldsymbol{c} \tag{A.32}$$

$$\widehat{\boldsymbol{a}}^T = -\widehat{\boldsymbol{a}} \tag{A.33}$$

$$\boldsymbol{a}, \boldsymbol{c} \in \mathbb{R}^n \tag{A.34}$$

B | UVMS Simulator

B.1 Simulation Parameters

B.1.1 Kinematics Parameters

The kinematics parameters of the manipulator is described by the following DH-table

Table B.1: DH parameters of the kinematic chain of the robot manipulator

Link	a_i	α_i	d_i	θ_i
1	0.15	$\frac{\pi}{2}$	0	q_1
2	1	0	0	q_2
3	0.7	0	0	q_3
4	0.4	$-\frac{\pi}{2}$	0	q_4
5	0	$-\frac{\pi}{2}$	0	q_5
6	0	0	0.25	q_6

B.2 About the Simulator Software

In order to simulate the UVMS dynamics, a simulator was created using Matlab/Simulink. Most of the simulator is written by the author, but some 3rd party software is also used. MSS GNC is a Matlab toolbox for guidance, navigation and control and is Copyrighted (C) 2008 Thor I. Fossen and Tristan Perez, and is licensed under GNU General Public License. The library functions in MSS GNC are used for some of the kinematic transformations of the system.

The Robotics, Vision And Control (RVC) Matlab toolbox is written by Peter Cork and is licensed under the GNU Lesser General Public License. The RVC toolbox is used for animation of the UVMS.

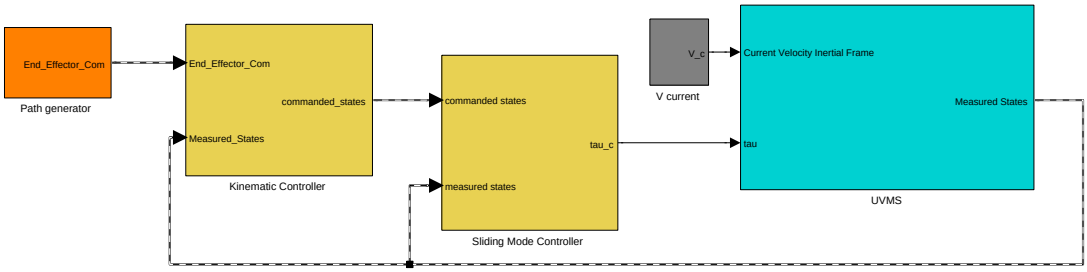


Figure B.1: UVMS Simulatoar

The UVMS simulator written by the author is licensed under the GNU General Public License, and can be downloaded from https://github.com/simena86/Simulink-Underwater-Robotics-Simulator. The Simulink diagram of the UVMS Simulator is illustrated in Fig. B.1. The dynamics and kinematics of the system is solved in the UVMS block of the diagram. The states $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ are then the output of the UVMS block. This output signal, named *measured states*, is implemented as a bus structure, which contains both the system states, as well as the kinematic transformations such as rotation matrices and Jacobians. This is to avoid calculating the same transformations in different places in the simulator. The kinematics and dynamics are based on the work done in this paper. The dynamics parameters are found simplifications based on modeling the structure as cylinders. It is, however, easy to modify both the dynamics and the DH-parameters, as this is scripted in separate files. The dynamics parameters can easily be changed by changing the inertial, hydrodynamic and persistent matrices for each rigid body.

B.3 A Quick Guide To the Simulator Software

A CD is attached containing the simulation software constructed for this paper. We will now show how to run a demo of the software.

B.3.1 Minimal Working Example

To initialize system path names for the current setup run the following in the Matlab terminal:

```
>>initUVMS;
```

It is very important that this file is run from the root folder, as it sets the paths of the system relative to the current directory.

To set how many seconds you want to simulate for, edit the file *uvms_ config.ini*. Then, to simulate the system for tracking prebuilt trajectories, run the following in the Matlab terminal

```
>>run_simulation;
```

This will initialize all the system parameters and run the Simulink target. Finally, to get plots or an animation from the simulation, run one of the two scripts:

```
>>animate_uvms;
% or
>>plot_uvms;
```

B.3.2 Generate End Effector Path

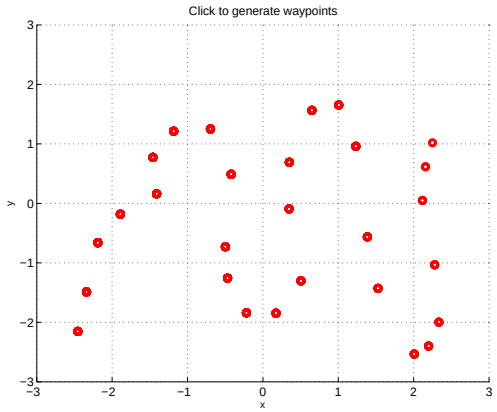
When running the simulator, without changing any parameters as described below, the end effector trajectory that is used for the simulation is located in the file *./data/trajectory.mat*. The trajectory can however be specified by a user, by following the following steps:

Step 1, create waypoints for translational motion of end effector:

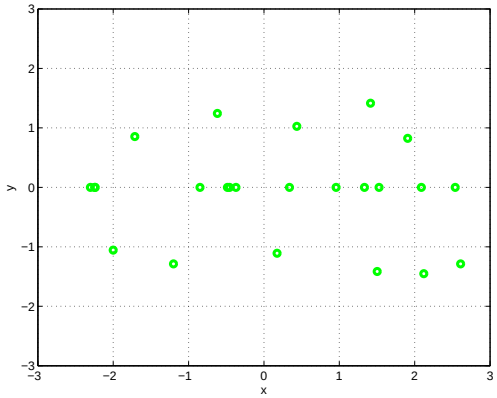
Run the following Matlab command (Make sure that initUVMS has been run in order to specify all paths).

```
>>makeWayPoints;
```

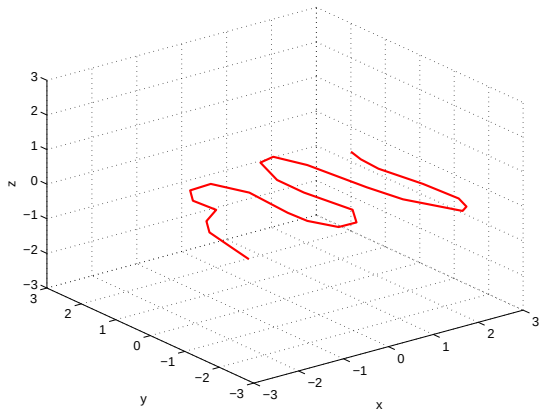
This command opens up two Matlab plot figures. The x and y coordinates of the waypoints can then be generated by clicking with the mouse pointer inside the first Matlab figure (see Fig. B.2a). By pressing the key *s* one can then specify the z coordinates of the generated points. This is done by dragging and dropping the points (see Fig. B.2b). The generated trajectory can be viewed in the 3D figure, see Fig. B.2c. Finally, press the key *q* to finish, and the waypoints are then saved in the *./data* folder.



(a) Waypoints are generated in the x-y plane, with mouse clicks



(b) Waypoints can be manipulated in the x-z plane



(c) 3D view of the generated trajectory

Step 2, create waypoints for rotational motion of end effector:

Run the following command matlab command (Make sure that initUVMS has been run in order to specify all paths).

```
>>makeOrientationWayPoints;
```

This will open up two windows, similar to the previous step. The roll and yaw angles of the manipulator can then be created (similar to the first window in the previous step). By pressing the *s* key one can then specify the pitch angle of the manipulator. When finished, press the *q* key, and the orientation waypoints will be saved to the *./data* folder.

Step 3, run scipt to interpolate waypoints:

The waypoints generated in the two previous steps can now be interpolated, using cubic splines, yielding smooth velocity trajectories. Run the following command:

```
>>timeInterpolate;
```

This will generate the interpolated trajectory. The velocity trajectory that corresponds to the generated waypoints are then saved as *trajectory.mat* in the *./data* folder. It should be noted that not all trajectories will give good tracking results. Waypoints that are too far away from each other will yield high velocities. High velocities, especially when the end effector is operating outside $\mathcal{W}_s$, will give bad results. This is because the vehicle's dynamics is slow, and it can therefore be difficult to follow the high velocity trajectory.

B.3.3 Changing Simulator Parameters

The different parameters of the simulator can easily be changed by editing the different initialization files. The initialization files for the simulator parameters are located in the *./uvms_functions* folder. Below are a short description of each initialization file:

- *init_kinematics.m* - DH-parameters, the joint limits and CG and CO of each rigid body of the UVMS.
- *init_kinetics.m* - rigid body parameters of each link, i.e. $\boldsymbol{M}_{i,RB}, \boldsymbol{M}_{i,A}, \boldsymbol{D}_i, \boldsymbol{C}_{i,RB}, \boldsymbol{C}_{i,A}$ and $\boldsymbol{N}_i$ for each link.
- *init_inputs.m* - the initial configuration of the UVMS and the ocean current $\boldsymbol{V}_c^0$
- *init_control.m* - the control parameters, and the parameters for the low-pass filters.

Bibliography

G. Antonelli. *Underwater Robots*. Springer Tracts in Advanced Robotics. Springer International Publishing, 2013. ISBN 9783319028767. URL http://books.google.no/books?id=t1VHngEACAAJ.

Tan Fung Chan and R.V. Dubey. A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *Robotics and Automation, IEEE Transactions on*, 11(2):286–292, 1995. ISSN 1042-296X. doi: 10.1109/70.370511.

Jack CK Chou. Quaternion kinematic and dynamic differential equations. *Robotics and Automation, IEEE Transactions on*, 8(1):53–64, 1992.

R. Featherstone. A beginner's guide to 6-d vectors (part 1). *Robotics Automation Magazine, IEEE*, 17(3):83–94, 2010. ISSN 1070-9932. doi: 10.1109/MRA.2010.937853.

O-E Fjellstad and Thor I Fossen. Singularity-free tracking of unmanned underwater vehicles in 6 dof. In *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, volume 2, pages 1128–1133. IEEE, 1994a.

O-E Fjellstad and Thor I Fossen. Position and attitude tracking of auv's: a quaternion feedback approach. *Oceanic Engineering, IEEE Journal of*, 19(4):512–518, 1994b.

O-E Fjellstad and Thor I Fossen. Singularity-free tracking of unmanned underwater vehicles in 6 dof. In *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, volume 2, pages 1128–1133. IEEE, 1994c.

T.I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011. ISBN 9781119998686. URL http://books.google.no/books?id=vCAzd3DaZCgC.

P.J. From, J.T. Gravdahl, and K.Y. Pettersen. *Vehicle-Manipulator Systems: Modeling for*

Simulation, Analysis, and Control. Advances in Industrial Control. Springer London, 2013. ISBN 9781447154624. URL `http://books.google.no/books?id=Ze1MngEACAAJ`.

Øyvind Hegrenæs. *Autonomous navigation for underwater vehicles*. PhD thesis, Norwegian University of Science and Technology, 2010.

Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.

H.K. Khalil. *Nonlinear Systems*. Prentice Hall PTR, 2002. ISBN 9780130673893. URL `http://books.google.no/books?id=t_d1QgAACAAJ`.

Jinyun Kim, Wan Kyun Chung, and Junku Yuh. Dynamic analysis and two-time scale control for underwater vehicle-manipulator systems. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 577–582 vol.1, 2003.

A. Liegeois. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Trans. Systems, Man, and Cybernetics*, 7(12):842–868, 1977.

Richard M. Murray, Zexiang Li, and S. Shankar Sastry. A mathematical introduction to robotic manipulation, 1994.

Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

I. Schjølberg and T. I. Fossen. Modelling and control of underwater vehicle-manipulator systems. In *in Proc. rd Conf. on Marine Craft maneuvering and control*, pages 45–57, 1994.

M.W. Spong and S. Hutchinson. *Robot Modeling and Control*. Wiley, 2005. ISBN 9780471649908. URL `http://books.google.no/books?id=wGapQAAACAAJ`.

Henry M Stommel. *The Gulf Stream: a physical and dynamical description*. Univ of California Press, 1958.