NTNU
Norwegian University of
Science and Technology

Master's thesis

2019

Jenny Aurora Frøland Steindal

Master's thesis

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

Jenny Aurora Frøland Steindal

Gain Scheduled Controller with Bumpless Transfer for an Unmanned Surface Vehicle

July 2019

NTNU
Norwegian University of
Science and Technology

NTNU

Norwegian University of
Science and Technology

Gain Scheduled Controller with Bumpless Transfer for an Unmanned Surface Vehicle

Jenny Aurora Frøland Steindal

Master of Science in Cybernetics and Robotics

Submission date:	July 2019
Supervisor:	Tor Arne Johansen
Co-supervisor:	Stephanie Kemna

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Acknowledgments

First I would like to thank my supervisors Tor Arne Johansen and Stephanie Kemna. Tor Arne has helped me to define the master thesis and has helped med when I have been stuck. Stephanie has helped me by having weekly meeting and pushed me to have a continuous work flow. She has also been a huge help in discussing possible solutions and helped me with the structural content of my thesis.

I am grateful for the hospitality that all in Maritime Robotics has given me. It was a pleasure sharing office with you. I would also thank my friend Sigurd for discussing multiple problems I have been facing during the work of my master thesis.

Last, but not least I would like to thank my big sister Inga for proofreading my thesis.

Executive Summary

This master thesis is based on the project report ”System identification of an unmanned surface vehicle with fixed thrusters”. The project report proposes a method to find the model parameters for an autopilot model for Maritime Robotics smallest unmanned surface vehicle, called the Otter. The autopilot model is a first order Nomoto model, which describes the relationship between the yaw rate and the input torque from the thrusters. Based on several different turning circles executed at different velocities two Nomoto models were found for low and high velocities.

The two Nomoto models are the foundation for this master thesis. The models are used to make a linear quadratic (LQ) controller for both high and low velocities. This is a nonlinear control method called gain scheduling, where the nonlinear system is linearised around several different equilibrium points. When the system is linearised, linear control theory can be used in the neighbourhood of the linearisation. The results are several linear controllers that can be switched between by using a variable, which captures the nonlinearities of the system. In this thesis the controllers are switched between by using the USV's velocity. To avoid sudden jumps in the controller output when the controllers are switching a bumpless transfer technique is used to prevent this.

The model that is used to simulate the Otter's dynamics is an interpolation of the model parameters for the two Nomoto models. The model parameters depends on the simulated velocity to the Otter USV.

All simulations in this master thesis are done in Matlab/Simulink. In addition I had access to acceleration tests executed by Maritime Robotics Fall 2018. This data was used to make an approximate model of the Otter USV's velocity.

The gain scheduled controller was compared to a static LQ controller to check if gain scheduling is a good control method for the Otter. The result was that the gain scheduled controller did not have a better performance than a static LQ controller. The models that were used in this thesis are based on the Otter being equipped with a sonar. This gives the Otter USV a velocity space of 0-2 m/s. There is a big uncertainty for the mathematical model of the Otter USV. The Otter is a nonlinear system, however, the simulations are based on a linear model, which are 100% the same as the linear models the controllers are based on. Furthermore, based on the results in can be concluded that a gain scheduled controller with bumpless transfer is not better than a static LQ controller.

Sammendrag

Denne masteroppgaven er basert på prosjektoppgaven “Systemidentifikajson av et ubemannet overflatefartøy med to faste propeller”. Prosjektoppgaven legger frem en metode for å finne modellparameterne til en autopilotmodell for Maritime Robotics sin minste overflatefarkost, kalt ”Otteren”. Autopilotmodellen er en første ordens Nomoto modell som beskriver sammenhengen mellom kursrate (yaw rate) og inputmomentet fra thrusterne. Basert på flere forskjellige “turning circles” gjort ved forskjellige hastigheter ble det funnet to Nomotomodeller for lav og høy hastighet.

Disse to Nomotomodellene danner grunnlaget for denne masteroppgaven. Modellene blir brukt til å lage en linear quadratic (LQ) kontroller for både høy og lav hastighet. Dette er en ulineær kontrollmetode kalt “gain scheduling”, der det ulineære systemet blir linearisert rundt diverse likevektspunkter og lineær kontrollteori blir brukt innenfor det lille området lineariseringene er gyldig for. Resultatet blir flere lineære kontrollere som må bli byttet i mellom ved hjelp av en variabel som fanger opp ulinæritetene til systemet. I denne oppgaven blir det byttet mellom de to kontrollerene basert på hastigheten til fartøyet. For å forhindre uønsket sprang i pådraget til båten når det skiftes mellom de to kontrollerne, har det også blitt implementert en “bumpless transfer”-teknikk for å forhindre dette.

Modellen som er brukt for å simulere dynamikken til Otteren, er en interpolering av modellparameterene til de to Nomotomodellene. Interpoleringen styres med hensyn på den simulerte hastigheten til Otteren.

Gain scheduled kontrolleren ble sammenliknet med en enkel LQ kontroller for å sjekke om gain scheduling er en god kontrollmetode for Otteren. Resultatet var at gain scheduled kontrolleren ikke har bedre ytelse enn en enkel LQ kontroller. Modellene som ble brukt i oppgaven baserer seg på at Otteren er utstyrt med en sonar. Dette gir Otteren et lite hastighetsområdet fra 0-2 m/s. Det ligger en stor usikkerhet rundt den matematiske modellen av Otteren. Otter USVen er egentlig et ulineært system, men simuleringene er basert på en lineær model som er 100% lik de linære modellene kontrollerne er basert på. Dette gjør det vanskelig å konkludere endelig.

Alle simuleringene i denne oppgaven er gjort i matlab/simulink, i tillegg har jeg fått tilgang på noen akselerasjonstester gjennom Maritime Robotics, som ble gjennomført høsten 2018. Denne dataen ble brukt til å lage en omtrentlig modell av hastigheten til overflatefarkosten.

Contents

List of Figures

List of Tables

Nomenclature

δ	Rudder angle
ψ	Yaw, heading
a	Moment arm for each thruster
F_i	Thrust from each propeller
K	Nomoto gain (system parameter)
Q_p	Applied torque from the thrusters. Input torque for the Nomoto model
r	Yaw rate
T	Nomoto time constant (system parameter)
U	Surge velocity
DOF	Degree of Freedom
IMU	Inertial measurement unit
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
SOG	Speed Over Ground
USV	Unmanned surface vehicle

Chapter 1

Introduction

1.1 Background

An unmanned surface vehicle (USV) is a vehicle that operates with close to continuous contact with the water surface. In 1898, Nikola Tesla patented a remote radio based control method for boats, which was the beginning of the USV. However, it was not exploited significantly until World War II, where USVs were used to lay smoke in the Normandy invasion, as well as for minesweeping operations and as targets for shooting practice for manned marine vessels (Breivik, 2010).

In Breivik (2010), the USV is described as a vessel that typically suits dirty, dull and dangerous missions:

- Dirty: e.g. operations in polluted areas. The USV was used to collect radioactive water samples at Bikini Atoll after detonation of an atomic bomb by the USA.
- Dull: e.g. seafloor mapping and maritime surveillance.
- Dangerous: e.g. military observations and minesweeping.

After World War II, the USV was mainly used in military operations as minesweeping boats. During this time, the focus shifted towards developing unmanned vehicles operating in the air or under the water surface, and consequently, the development of the USV was not prioritised. Fast forwarding to the beginning of the 90s, there was a newly established interest of developing advanced technology for the USV. As a result, there has been a strong increase in demands to develop USVs for military, commercial and scientific groups due to national security issues, personnel requirements and climate change (Liu et al., 2016).

Maritime Robotics was founded in 2005 and makes vehicle control systems and sensors for unmanned vehicles. One of their USVs is the Otter, which is the smallest USV produced by Maritime Robotics. It can be used for seabed mapping and monitoring of sheltered waters (Maritime Robotics, 2019). The Otter has a catamaran hull, measuring $2.00 \times 1.05 \times 0.85$ meters, and is equipped with two fixed electric propellers, which means that turning the vessel is executed by the different forces form the propellers. In this master thesis, I will propose a controller that depends on the vessel's velocity to obtain precise tracking of the heading angle.

System description

The Otter USV is a catamaran style vehicle and is directionally stable. The Otter USV is equipped with a GPS (global positioning system) and an IMU (inertial measurement unit). The GPS measures the vessel's position, course and speed over ground. The IMU measures the specific force, angular rate, and the heading of the USV.

The Nomoto model is used to describe the heading dynamics of the Otter USV, which has the following states: yaw (ψ) and yaw rate (r). In this master thesis full state feedback is used as the control method. Since all of the states are measured, there is no need for an observer.

Problem description

In my project report (Steindal, 2018), I proposed two different first order Nomoto models for the Otter USV. The autopilot models depended on the velocity of the vessel, so the result was one Nomoto model for slow turning and one model for fast turning. A schematic description of the work discussed in this thesis is depicted in Figure 1.1. In the thesis these two models will be converted into a state space model and used to make two linear quadratic regulators (LQR). The two controllers are tuned independently for its corresponding autopilot model. This method is know as gain scheduling and is a widely used nonlinear control method in the industry. Furthermore, the result of this is two controllers that can be switched between when the USV hits a certain velocity. The consequence of switching between different control parameters is that the controller output will make a sudden jump. This should be avoided, so a bumpless transfer method is proposed to maintain a smooth input for the system.

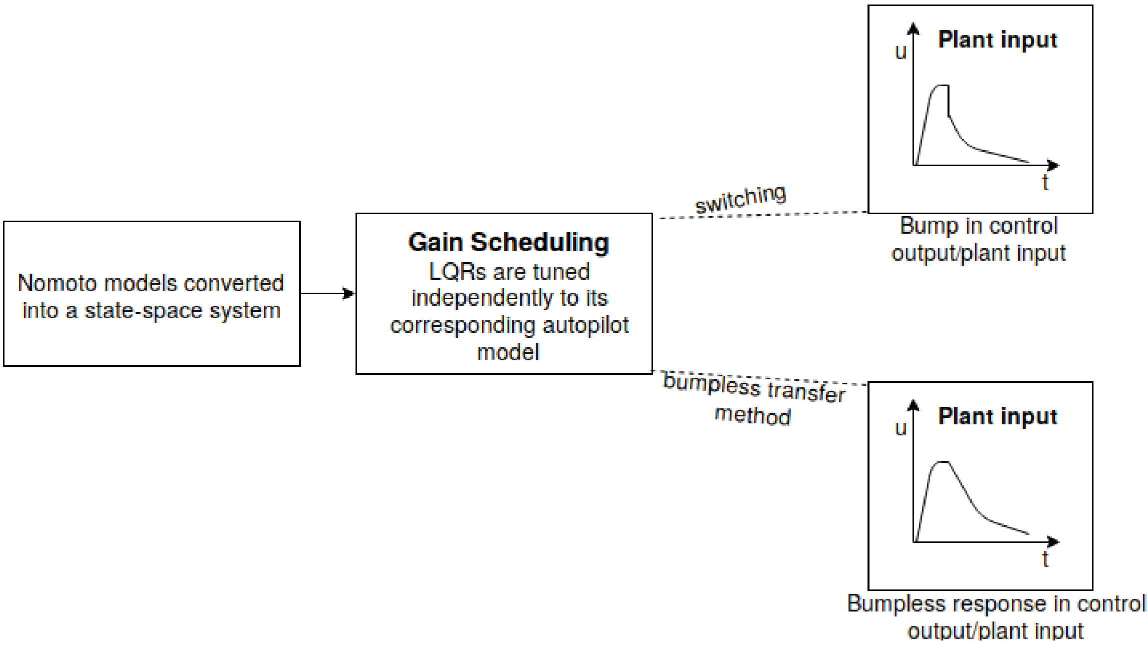

Nomoto models converted into a state-space system
Gain Scheduling
LQRs are tuned independently to its corresponding autopilot model
switching
bumpless transfer method
Plant input
u
t
Bump in control output/plant input
Plant input
u
t
Bumpless response in control output/plant input

Figure 1.1: Overview of the steps to obtain a heading controller.

Objectives

- Convert the Nomoto model to state-space system and add integral action as an augmented state.
- Tune each LQR controller for best response in slow and fast turning (gain scheduling).
- Implement a bumpless transfer technique to avoid sudden jumps when the velocity changes.
- Find an approximation of the surge dynamics to be used in the simulator as the scheduling variable.

Limitations

The simulated model of the Otter USV is equal to the Nomoto models that the LQR is based on. A non-linear model of the USV would have given more realistic results. When performing acceleration tests, an error in the logging caused a loss of velocity data, hence the surge model of the Otter USV is based only on a partial data.

1.2 Outline

Chapter 2

This chapter describes the theory behind the the Nomoto model, linear quadratic regulator (LQR), gain scheduling and bumpless transfer. It also describes the effect of integral windup, when the system has saturation limits on the input.

Chapter 3

This chapter describes first how the models of the Otter USV are implemented in Simulink and how the parameters are interpolated. Secondly the implementation of the LQR with gain scheduling is described. At last the actuator limits are obtained and the anti-windup scheme is described.

Chapter 4

This chapter presents the results for thrust curve of the actuators, the tuning of the two LQ controllers, the effect of bumpless transfer and a comparison of the gain scheduled LQ controller with a static LQ controller. The results are discussed and possible explanations for the obtained results are proposed.

Chapter 2

Theory

The purpose of this chapter is to describe the theory behind a gain scheduled controller for Maritime Robotics's smallest USV, the Otter. The Otter USV can be described mathematically by non-linear equations, the set of equations describing the Otter may be referred to as the system. A non-linear system can be linearised around equilibrium points. However, the linearisation is only valid within a small neighbourhood around the linearisation point. Furthermore, by dividing the non-linear system into multiple operating points, and then linearising around these, the non-linear system will be approximated by multiple linear models. For each linear model, a linear controller is designed to stabilise the linear system.

In this thesis the system is the unmanned surface vehicle, the Otter, developed by Maritime Robotics, as described in the introduction. This chapter will start by presenting a non-linear model for the USV, before I describe how the linearisation of the model will be derived. Based on this linearisation, a 1 degree of freedom autopilot model will be derived.

The 1 degree of freedom (DOF) autopilot model, also known as the Nomoto model, will be used as the linear time invariant model, which will be used when designing a linear quadratic regulator (LQR) for each linear model. A state space representation of the Nomoto model will be presented and analysed. This chapter will end with a short review of different gain scheduling methods. Since switching between controllers with different control parameters can cause jumps in the input, a bumpless transfer method is presented to avoid this.

The different vessel models and their relationships are summarised in Figure 2.1

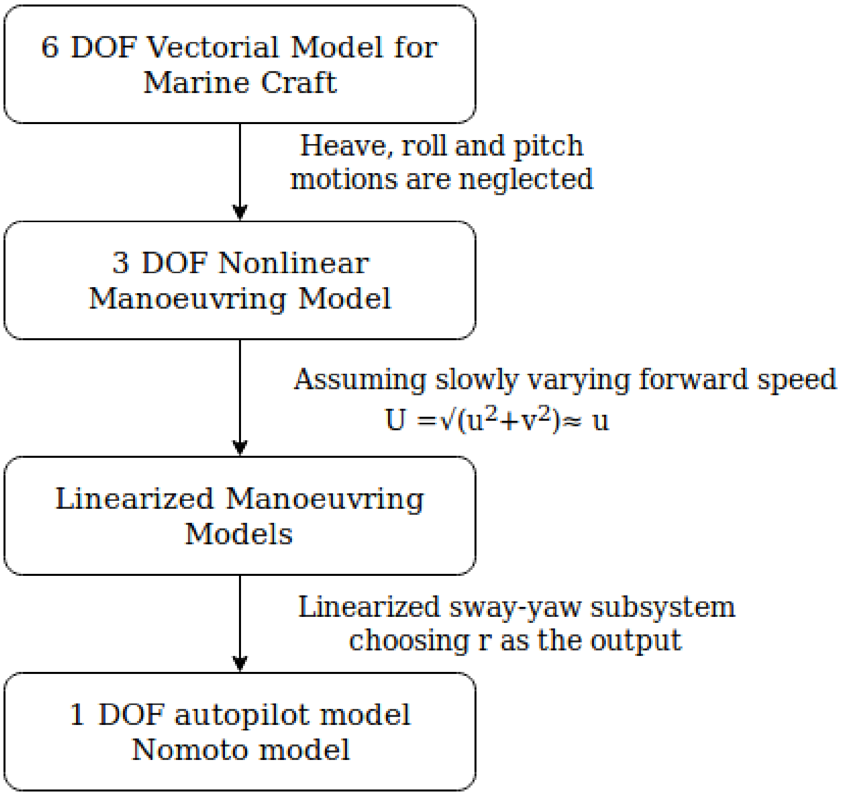

6 DOF Vectorial Model for Marine Craft
Heave, roll and pitch motions are neglected
3 DOF Nonlinear Manoeuvring Model
Assuming slowly varying forward speed
U =√(u²+v²)≈ u
Linearized Manoeuvring Models
Linearized sway-yaw subsystem choosing r as the output
1 DOF autopilot model Nomoto model

Figure 2.1: Summary of all the vessel models, which will be described more thoroughly in the next sections.

2.1 Vessel Model

The Otter USV is a surface vehicle, which means it moves in six degrees of freedom (DOFs). The position and orientation of the Otter is determined by six independent coordinates, which are depicted in Figure 2.2. The horizontal movement of a surface vessel can be described by a 3 DOF manoeuvring model. The derivation of the 3 DOF manoeuvring model is done in Section 2.1.2.

The objective of this thesis is to implement a gain scheduled controller for the Otter. Two linear quadratic regulators (LQR) are tuned based on two Nomoto models proposed in Steindal (2018). The Nomoto models are trying to capture the dynamics, which change with the velocity. The Nomoto model is a 1 DOF autopilot model, which describes the heading dynamics. The heading, ψ, is defined as depicted in Figure 2.3.

The Otter USV has two fixed electric thrusters located aft. The vessel turns when there is a thrust difference. The control inputs are

$$\begin{aligned} u_1 &= Q_p = (F_{port} - F_{stb})a \quad [\text{Nm}] \\ u_2 &= T_p = F_{port} + F_{stb} \quad [\text{N}] \end{aligned} \tag{2.1}$$

where Q_p is the propeller torque and T_p is the total thrust, F is the thrust from each propeller, a is the moment arm, and u_1 is the input to the Nomoto model, while u_2 is the input to a surge model.

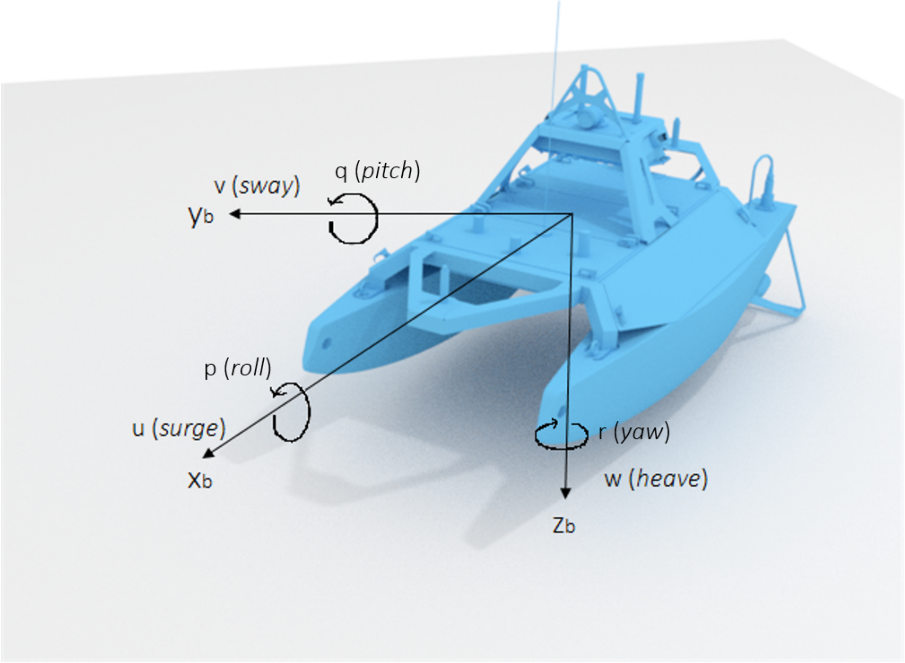


Figure 2.2: The Otter with 6 DOF velocities in body-fixed reference frame

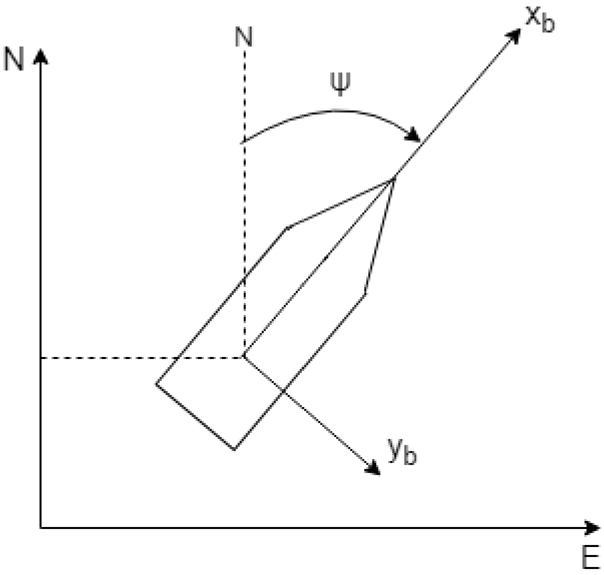


Figure 2.3: Planar motion of the Otter

2.1.1 6 DOF Vectorial Model for Marine Craft

The 6 DOF model for a marine craft can be written as (Fossen, 2011)

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu} \tag{2.2a}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) + \mathbf{g}_0 = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \tag{2.2b}$$

where

$$\boldsymbol{\eta} = \left[x, y, z, \phi, \theta, \psi\right]^\top \tag{2.3}$$

$$\boldsymbol{\nu} = \left[u, v, w, p, q, r\right]^\top \tag{2.4}$$

Vector $\boldsymbol{\eta}$ consists of position and Euler angles, and $\boldsymbol{\nu}$ consists of the velocities. These are summarised in Table 2.1. These are necessary to describe the motions in 6 DOF, which can be seen in Figure 2.2. The model matrices, $\mathbf{M}$, $\mathbf{C}(\boldsymbol{\nu})$ and $\mathbf{D}(\boldsymbol{\nu})$ denote inertia, Coriolis and damping. The generalised buoyancy and gravitational forces are collected in $\mathbf{g}(\boldsymbol{\eta})$. The static restoring forces and moments due to ballast are found in $\mathbf{g}_0$. The generalised forces acting on the craft are collected in $\boldsymbol{\tau}$.

Table 2.1: The notation of SNAME (1950) for marine vessels, where b-frame is the Otter USV's body frame, n-frame is the north-east-down reference frame.

DOF		Forces and moments (b-frame)	Linear and angular velocities (b-frame)	Positions and Euler angles (n-frame)
1	motions in x direction (surge)	X	u	x
2	motions in y direction (sway)	Y	v	y
3	motions in z direction (heave)	Z	w	z
4	rotation around x axis (roll, heel)	K	p	ϕ
5	rotation around y axis (pitch, trim)	M	q	θ
6	rotation around z axis (yaw)	N	r	ψ

2.1.2 3 DOF Nonlinear Manoeuvring Model

Manoeuvring theory assumes that the vessel is moving with constant positive speed U in a zero-frequency wave excitation. This means that added mass and damping can be expressed by constant parameters. The zero-frequency assumption is only valid for surge, sway and yaw (Fossen, 2011). Hence the heave, roll and pitch motions are neglected and the 3 DOF manoeuvring model can be written as a coupled surge-sway-yaw model (Fossen, 2011)

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{2.5}$$

where $\mathbf{R}(\psi)$ is the transformation matrix from body frame to NED frame

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

Introducing the relative velocity vector

$$\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c \tag{2.7}$$

where the irrotational ocean current velocity can be written as

$$\boldsymbol{\nu}_c = \left[u_c, v_c, w_c, 0, 0, 0\right]^\top \tag{2.8}$$

The manoeuvring equations of motion can be written as

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{N}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave} \tag{2.9}$$

and the term $\mathbf{N}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r$ has the following expression

$$\mathbf{N}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r := \mathbf{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}\boldsymbol{\nu}_r + \mathbf{d}(\boldsymbol{\nu}_r) \tag{2.10}$$

The linear damper $\mathbf{D}$ dominates for low-speed manoeuvring, while the cross-flow drag matrix $\mathbf{d}(\boldsymbol{\nu}_r)$ is important for higher speeds. The linear damping is also important because it assures the velocity to converge exponentially to zero (Fossen, 2011).

The vectors consist of $\boldsymbol{\nu} = \left[u, v, r\right]^\top$ and $\boldsymbol{\eta} = \left[N, E, \psi\right]^\top$. It is also assumed that the vessel has homogeneous mass distribution and xz- plane symmetry ($I_{xy} = I_{yz} = 0$).

A homogeneous mass distribution means that the vessel has a constant density, which is a reasonable assumption for the Otter USV, since the mass and volume of the Otter is constant. xz-plane symmetry assumes that the Otter USV is symmetric around the xz-plane (body-frame). This is also a reasonable assumption, which can be studied in Figure 2.2. These two assumptions simplify the model equations.

Many physical systems can be approximated by a linear system. Linear systems has the superposition property, which means that the effect of two or more simultaneous stimuli is the sum of the effects that would have been caused by each stimulus individually. In physics and engineering this principle has many applications. A common assumption in hydrodynamics is that hydrodynamic forces and moments acting on a rigid body can be linearly superimposed. Based on this, the manoeuvring equations can be looked at as a mass-damper-spring system (Fossen, 2011).

2.1.3 Linearized Manoeuvring Models

A marine vessel is moving with a slowly varying forward speed (Fossen, 2011):

$$U = \sqrt{u^2 + v^2} \approx u \tag{2.11}$$

The manoeuvring model from Equation 2.9 can be decoupled in a surge (forward speed) subsystem and a sway-yaw subsystem, because of port-starboard symmetry.

The surge subsystem can be written as

$$(m - X_{\dot{u}})\dot{u} - X_u u_r - X_{|u|u}|u_r|u_r = \tau_1 \tag{2.12}$$

where $X_{|u|u}|u_r|u_r$ is the damping term.

The sway-yaw subsystem can be written as

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{N}(u_0)\boldsymbol{\nu}_r = \mathbf{b}\delta \tag{2.13}$$

where δ is the rudder angle and $\boldsymbol{\nu} = [v, r]^\top$, where v is the sway velocity and r is the yaw rate. This model is based on the assumptions that v and r are small and that the cruise speed u is approximately constant.

$$u = u_0 \approx \text{constant} \tag{2.14}$$

2.1.4 Nomoto Model

The Nomoto model is a 1 DOF autopilot model proposed by Nomoto et al. (1957). It is a widely used autopilot model, which can be derived from the linearized sway-yaw subsystem as defined in Equation 2.13. The Nomoto model can be derived in the following way (Fossen, 2011)

By choosing r as the output

$$r = \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{\nu} \tag{2.15}$$

Applying the Laplace transform, the second order Nomoto model yields

$$\frac{r}{\delta}(s) = \frac{K(1 + T_3 s)}{(1 + T_1 s)(1 + T_2 s)} \tag{2.16}$$

The second order model can be reduced to a first order Nomoto model by defining a new time constant as

$$T := T_1 + T_2 - T_3 \tag{2.17}$$

The first order Nomoto model then becomes

$$\frac{r}{\delta}(s) = \frac{K}{Ts + 1} \tag{2.18}$$

Using $\dot{\psi} = r$ the obtained result for a heading autopilot model is

$$\frac{\psi}{\delta} = \frac{K}{s(Ts + 1)} \tag{2.19}$$

which can be written in the time domain as

$$T\ddot{\psi} + \dot{\psi} = K\delta \tag{2.20}$$

Nomoto (1960) explained the parameter K as the vessel's ability to turn. The larger the K value of a vessel, the greater the angular turning rate, r, and subsequently a smaller steady turning circle.

T holds information on the vessel's course stability and quick response in steering. The speed with which a vessel can approach the terminal angular rate is determined by T.

The smaller the T, the quicker build-up of the angular turning rate.

2.1.5 State Space

For further investigations of the proposed Nomoto models, a structural analysis should be made. Since the Nomoto model is a linear system, the two equations can be described in state space form. The advantage of the state space is the structural analysis that can be done using the matrices defining the USV system. These analyses give valuable understanding of the Otter USV, which will be important when designing a controller for the USV.

A linear time varying lumped system can be described by

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \tag{2.21a}$$

$$\mathbf{y} = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \tag{2.21b}$$

Equation 2.21a is a set of linear differential equations, where $\mathbf{x}$ is the state vector, $\mathbf{A}(t)$ is the state matrix and $\mathbf{B}(t)$ is the input matrix. These equations describe the dynamics of the linear system. While 2.21b is a set of algebraic equations, where y is the output, $\mathbf{C}(t)$ is the output matrix and $\mathbf{D}(t)$ is the feedforward matrix (Chen, 1999).

However, since the Nomoto model is a linear time invariant system, meaning

$$y = \int_0^t \mathbf{G}(t-\tau)\mathbf{u}(\tau)d\tau \tag{2.22}$$

the Equation 2.21 reduces to

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y} &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{2.23}$$

Equation 2.20 can be written in state space form, where the state vector $\mathbf{x} = \begin{bmatrix}\psi & r\end{bmatrix}^\top$. This gives

$$\begin{bmatrix}\dot{\psi} \\ \dot{r}\end{bmatrix} = \begin{bmatrix}0 & 1 \\ 0 & -\frac{1}{T}\end{bmatrix}\begin{bmatrix}\psi \\ r\end{bmatrix} + \begin{bmatrix}0 \\ \frac{K}{T}\end{bmatrix} u \tag{2.24a}$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\psi \\ r\end{bmatrix} \tag{2.24b}$$

2.1.6 Controllability and Observability

In order to be able to design a linear optimal control system, Equation 2.23 has to be controllable, but also observable in case some of the states need to be estimated (Fossen, 2011).

Observability for Linear Time-Invariant Systems

Definition 1 *The state Equation 2.23 is said to be observable if for any unknown initial state* $\mathbf{x}(0)$*, there exists a finite* $t_1 > 0$ *such that the knowledge of the input* $\mathbf{u}$ *and the output* $\mathbf{y}$ *over* $\begin{bmatrix} 0 & t_1 \end{bmatrix}$ *suffices to determine uniquely the initial state* $\mathbf{x}(0)$*. Otherwise, the equation is said to be unobservable. (Chen, 1999)*

The observability matrix is defined as

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{2.25}$$

Controllability for Linear Time-Invariant Systems

Definition 2 *The pair* $(\mathbf{A}, \mathbf{B})$ *is said to be controllable if for any initial state* $\mathbf{x}(0) = \mathbf{x}_0$ *and any final state* $\mathbf{x}_1$*, there exists an input that transfers* $\mathbf{x}_0$ *to* $\mathbf{x}_1$ *in a finite time. Otherwise 2.23 is said to be uncontrollable. (Chen, 1999)*

The controllability matrix, $\mathcal{C}$, is:

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \dots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix} \tag{2.26}$$

If the controllability matrix $\mathcal{C}$ (Eq. 2.26) has full row rank, then state Equation 2.23 is controllable.

2.2 Linear Quadratic Regulator

Optimal control focuses on finding a control law for a given system to satisfy an optimality criterion. The optimality criterion is often a cost function J, which depends on the system's states and input variables.

For designing an optimal controller for the Otter USV, the control objective is to regulate the yaw angle to follow a desired path. To achieve the control objective, different linear control techniques can be done. A PID controller is an easy controller that uses the error between the measured process variable and desired reference. However the control parameters can sometimes be tricky to tune, and if there is a delay in measurements this can cause problems. Pole-placement is a state feedback controller, where a feedback gain matrix is calculated based on the designer's choice of the locations of the the poles of the system. A LQR controller is an optimal controller where a feedback gain is calculated based on the Riccti equation. The advantage of the LQR compared to the pole-placement controller is that it has an optimal and a more intuitive way of tuning the control parameters. The different properties of the controllers are summarised in the following list

- PID controller
  - the error of the system is corrected with a proportional, integral and derivative terms.
  - widely used in the industry.
  - can be tricky to tune. Trial and error.
  - delay in measurements of the process variable can cause problems.
- Pole placement
  - state feedback controller
  - the closed-loop poles of the system are placed in pre-determined locations by the designer.
  - system must be controllable
- LQR
  - optimal controller
  - optimal way of finding feedback gains
  - system must be controllable
  - more intuitive tuning than pole placement
  - based on a linear model of the system

Furthermore, to be able to use a LQR controller, the system has to be controllable and linear time invariant, which is not that common in the physical world. However, the Nomoto model is linear and time invariant, which makes the the LQR controller a good choice for the Otter USV. The LQR controller is a robust controller that achieves infinite gain margin. It also guarantees a phase margin $\gamma \geq 60°$ (Triantafyllou and Hover, 2003). A closed-loop system is stable, if the gain margin and the phase margin is positive (Kim, 2017). A PID controller could be a good option to the LQR controller in cases where the system does not fulfil the information needed to make a LQR controller. The advantage with the PID controller is the more dynamical control procedure since it depends on the error between the set point and the output. The problem can be to tune the PID controller, however, the control parameters can be found optimally by calculating the feedback gain matrix of the LQR, which will give the optimally gains for the PID controller.

However, since all states of the the heading system can be measured on the Otter USV, the LQR is used as the controller.

The cost function J can be written as (Fossen, 2011)

$$\begin{aligned} J &= min\Big\{\frac{1}{2}\int_0^T (\mathbf{y}^\top \mathbf{Q}\mathbf{y} + \mathbf{u}^\top \mathbf{R}\mathbf{u})\, dt\Big\} \\ &= min\Big\{\frac{1}{2}\int_0^T (\mathbf{x}^\top \mathbf{C}^\top \mathbf{Q}\mathbf{C}\mathbf{x} + \mathbf{u}^\top \mathbf{R}\mathbf{u})\, dt\Big\} \end{aligned} \tag{2.27}$$

where $\mathbf{Q} = \mathbf{Q}^\top \geq 0$ and $\mathbf{R} = \mathbf{R}^\top > 0$ are the weighting matrices. The steady-state solution for this problem is (Athans and Falb, 1966)

$$\mathbf{u} = -\underbrace{\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P}_\infty}_{\mathbf{K_c}} \mathbf{x} \tag{2.28}$$

where $\mathbf{K_c}$ is the feedback gain matrix and $\mathbf{P}_\infty$ is found by solving the Riccati equation

$$\mathbf{P}_\infty \mathbf{A} + \mathbf{A}^\top \mathbf{P}_\infty - \mathbf{P}_\infty \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P}_\infty + \mathbf{C}^\top \mathbf{Q}\mathbf{C} = \mathbf{0} \tag{2.29}$$

Trajectory tracking for the Otter USV is obtained by setting the input equal to

$$u = \psi_{ref} - \mathbf{K_c}\psi \tag{2.30}$$

The state feedback loop is illustrated in Figure 2.4.

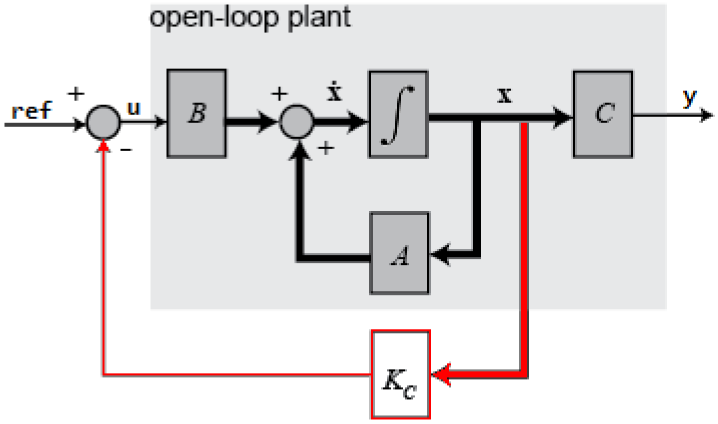


Figure 2.4: Trajectory tracking with LQR control. Courtesy of Messner et al. (2017)

There are two types of control; open loop or closed loop. If the input signal, $u(t)$, only depends on the reference signal and is independent of the output signal, the system is controlled in open loop. However, if the input signal depends on the reference signal and the output signal. Figure 2.4 depicts the reference signal as `ref` and the output signal as `y`, the system is controlled by closed loop or feedback control. Since the Otter USV is affected by noise and disturbances, open loop control will not work in a satisfactory manner. A feedback system will however manage to suppress disturbance and noise (Chen, 1999).

2.2.1 Integral Action

For the controller to handle constant disturbances, such as for example water current, an integral term can be added to the controller. The controller now consist of a proportional gain and a derivative gain which are located in $\mathbf{K}_c$ and a integrator term depicted as K_i in Figure 2.5 . This is very similar to the PID-controller. However, for the LQR controller, gains are found optimally by solving the Riccati equation, tuning the $\mathbf{Q}$ and $\mathbf{R}$ matrices. Furthermore, the block diagram, Figure 2.5, shows the concept of the integral action.

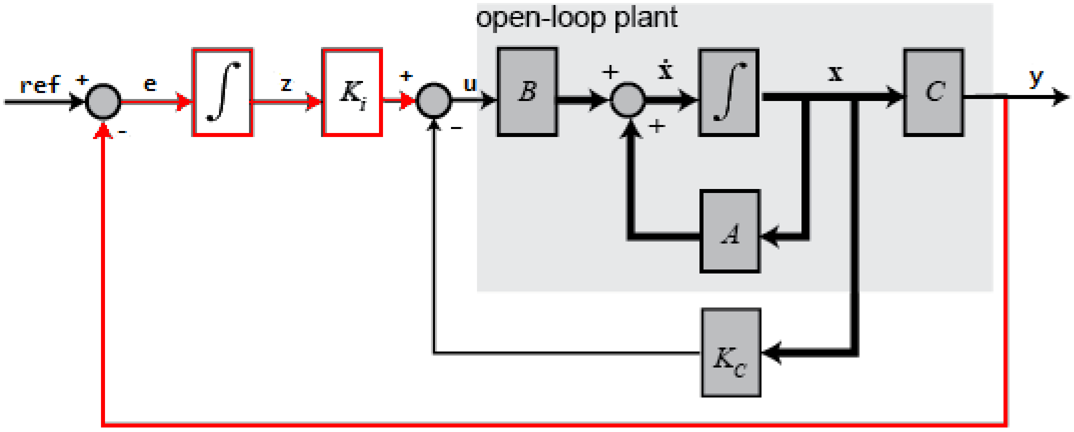


Figure 2.5: Block diagram of the system with integral action. Courtesy of Messner et al. (2017)

.

Furthermore, the integral action can be augmented in the state space system. Fossen (2011) describes this by defining a new state, the error between yaw and desired yaw angle

$$\dot{z} = \psi - \psi_{ref} = e \tag{2.31}$$

this gives

$$z = \int_0^t e(\tau) d\tau \tag{2.32}$$

obtaining the following state space model

$$\begin{bmatrix} \dot{z} \\ \dot{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{C} \\ 0 & \mathbf{A} \end{bmatrix} \begin{bmatrix} z \\ \mathbf{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{B} \end{bmatrix} u \tag{2.33}$$

$$y = \begin{bmatrix} 0 & \mathbf{C} \end{bmatrix} \begin{bmatrix} z \\ \mathbf{x} \end{bmatrix} \tag{2.34}$$

The state feedback becomes

$$u = -\begin{bmatrix} K_i & \mathbf{K_c} \end{bmatrix} \begin{bmatrix} z \\ \mathbf{x} \end{bmatrix} \tag{2.35}$$

where K_i is the integral gain and $\mathbf{K_c}$ contains the proportional gain and derivative gain.

2.2.2 Weighting Matrices

The main idea in LQR control design is to minimise the quadratic cost function J shown in Equation 2.27. Regardless of the values of $\mathbf{Q}$ and $\mathbf{R}$, this cost function has a unique minimum that can be obtained by solving the algebraic Riccati Equation (2.29).

The parameters $\mathbf{Q}$ and $\mathbf{R}$ can be used as design parameters to penalise the state variables and the control signals. The larger these values are, the more you penalise these signals. By choosing a large value for $\mathbf{R}$, you try to stabilise the system with less (weighted) energy. This is called an *expensive control strategy*. On the other hand, choosing a small value for $\mathbf{R}$ will not penalise the control signal, a *cheap control strategy*. Similarly, if a large value is chosen for $\mathbf{Q}$, the controller tries to stabilise the system with the least possible changes in the states, and a small value for $\mathbf{Q}$ implies less concern about the changes in the states (Triantafyllou and Hover, 2003).

The control objective is to have the best possible tracking for the heading. A large $\mathbf{Q}$ value for the heading state is a good start. Smaller values for $\mathbf{R}$ will lead to a large control signal, which can introduce sensor noise, or saturate the actuators.

2.2.3 Integral Windup

All physical actuators have a limit in their performance. This is an important aspect to think about when designing a controller with integral action. For the Otter USV the two electrical propellers are limited by a certain amount of thrust. This limitation will give a maximum torque, which is the input to the Nomoto model.

The torque input on the Otter USV takes the following form

$$u(t) = \begin{cases} u_{max}, & u > u_{max} \\ u(t), & -u_{max} \leq u \leq u_{max} \\ -u_{max}, & u < -u_{max} \end{cases} \tag{2.36}$$

Integral windup occurs when the input is saturated. The integrator keeps integrating up the error, once the error decreases the large integrator value prevents the controller to settle fast because the integral error has to decrease first. This gives a delayed response, which is not acceptable (Frazzoli, 2010).

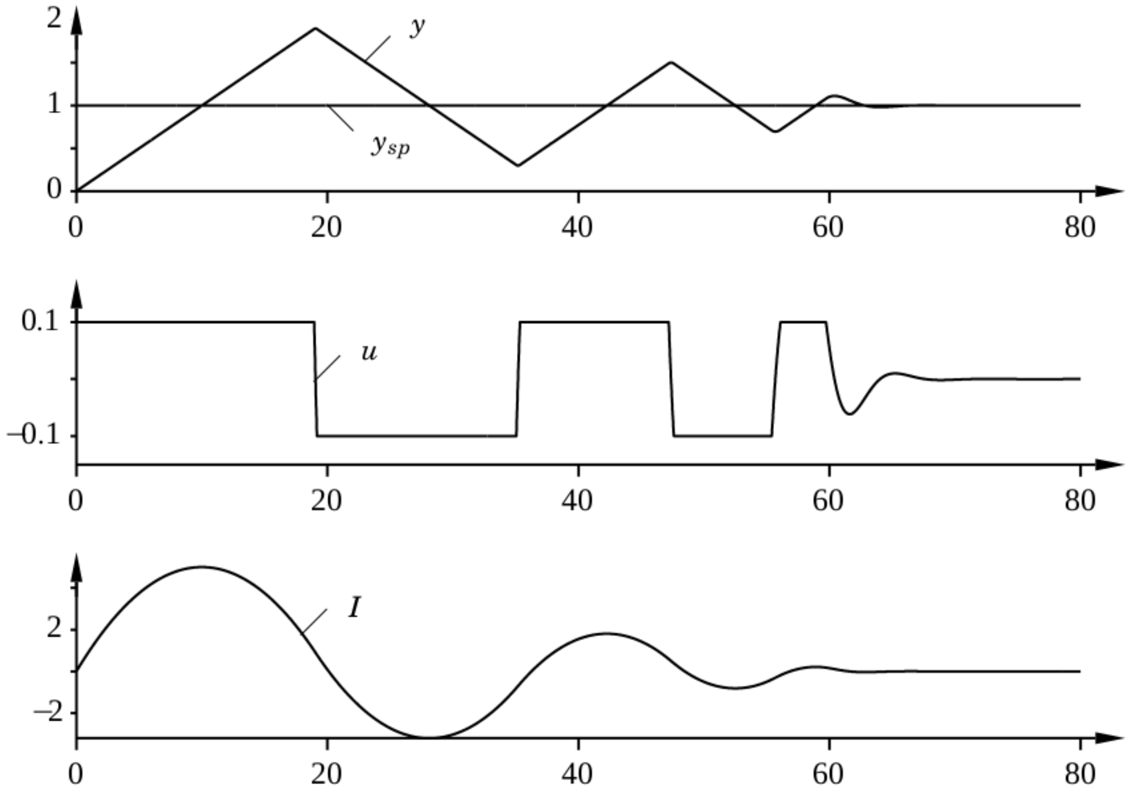


Figure 2.6: Illustration of an integral windup. Courtesy of Åström (2002)

However, anti-windup techniques can be used to prevent this phenomena. One option is to use a method that switches the integrator on and off depending on if the input is saturated or not. Åström and Rundqwist (1989) called this method the incremental algorithm and it originates from when controllers were implemented with analogue techniques. In the beginning of feedback control the actuator and the integral action were combined by having a motor drive the faucet directly. In this case windup was avoided, because the integration stops when the valve stops. When controllers were implemented with computers, the analog method was translated directly, leading to the incremental algorithm.

From Frazzoli (2010) the incremental algorithm can be written as

$$K_i' = \begin{cases} K_i & \text{if input is does not saturate} \\ 0 & \text{if input is saturated} \end{cases} \tag{2.37}$$

Back-calculation is another anti-windup method that compares the actual input with the saturated input. If they are the same the input error is zero and will not affect the system in normal operation. However, if the input is saturated Equation 2.38 reduces the integral error by a constant, times the difference between the input and the saturated input (Frazzoli, 2010).

$$u_{int} = \int K_i(e + T_{aw}(sat(u) - u))dt \tag{2.38}$$

The advantage with the back-calculation method is that the time constant, T_{aw}, changes the integral action dynamically and not in one time step as the incremental algorithm does (Åström and Rundqwist, 1989).

2.3 Gain Scheduling

When a controller is designed based on a linearized system the controller is guaranteed to only work in some neighbourhood of the equilibrium point. A solution to this is to use a technique that can enhance the validity of the linearisation approach to a range of operating points. This approach is called gain scheduling, and originated from the work of flight control systems (Khalil, 2001).

Gain scheduling is an approach to control nonlinear systems by using a family of linear controllers. Each linear controller is providing a satisfactory control for its specific operating point of the system. The operating points are described by a scheduling variable. The scheduling variable adjusts the controller gains based on the operating points. Figure 2.8 depicts a basic setup of a gain scheduled controller.

Since each linear controller can be tuned independently, the structure of the gain scheduling control method can be looked at as a divide and conquer approach for controlling a nonlinear system. Another advantage is that well established linear control theory can be used to control a nonlinear system.

It is not required that the scheduling variable is continuous. An apt scheduling variable is chosen based on physical insight. Gain scheduling is most suitable if the control bandwidth is faster than the gain scheduling variable (Leith and Leithead, 2000).

In Steindal (2018), the Otter USV heading dynamics was fitted to the Nomoto model, Equation 2.20, furthermore it was concluded that the heading dynamics depended on the vessel's velocity. Therefore, one Nomoto model was proposed for slow velocities and one for fast velocities. The plan is to use these two linear models to make a linear quadratic regulator (LQR), which are tuned to satisfaction to each of the model's operating point. The GPS on the Otter measures speed over ground (SOG), and by using it as the gain scheduling variable the two controllers can be switched between and give a better response than a single LQR. This will be presented in Chapter 3.

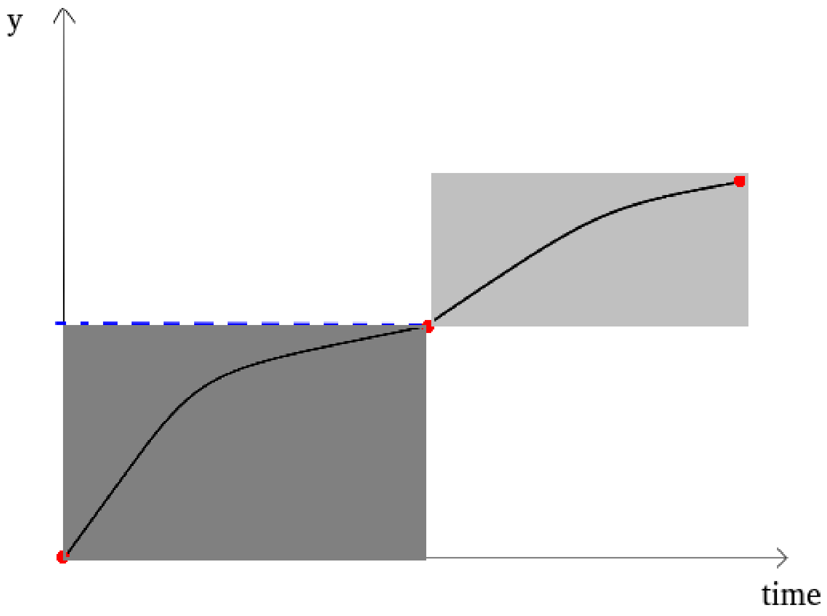


Figure 2.7: Illustration of gain scheduling. The red dots are operating points. Each point has a controller tuned independently for each point. The grey areas have a gain scheduled controller allowing for scheduling between the two operating points. Courtesy of Bendtsen et al. (2005)

Shamma and Athans (1990) describe the following design procedure for nonlinear systems

Step 1: Choose several operating points which cover the range of the system's dynamics.

Step 2: At each of these operating points, a linear time-invariant model is made for the system, and a controller is made for each linearised system.

Step 3: In between operating points, the gains of the controller are interpolated, or scheduled, thus resulting in a global controller.

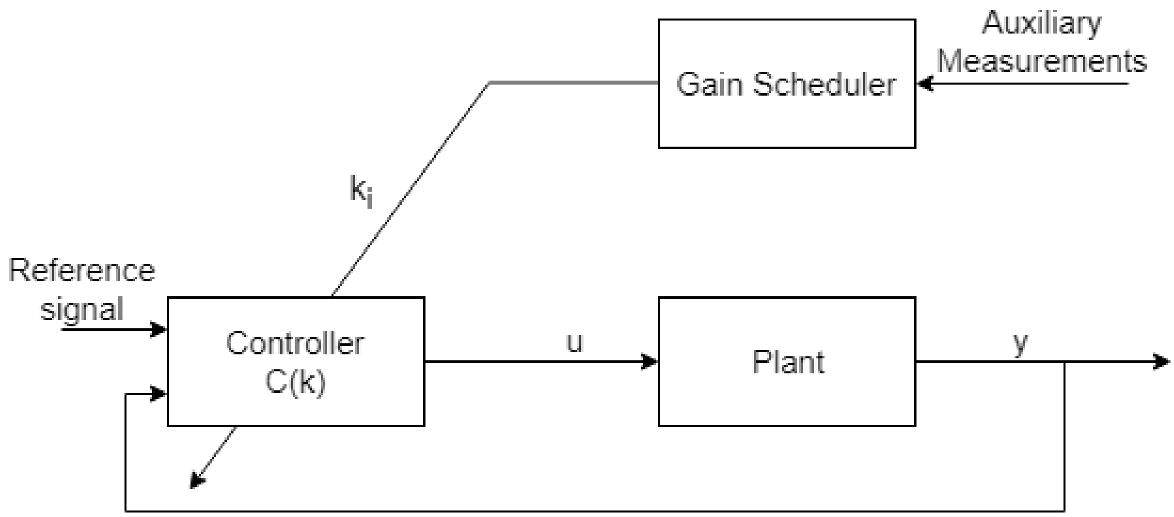


Figure 2.8: Basic gain scheduling scheme. Courtesy of Arpaia et al. (2014)

2.4 Bumpless Transfer

In gain scheduling each operating point has a LTI approximation, which is valid in a small region around the equilibrium point. These have a corresponding linear controller, which is tuned independently for each operating point. The scheduling variable decides when the systems goes from one operating point to another, which switches to a new controller with different controller gains.

For the Otter USV, a LQR controller is designed based on the slow Nomoto model and another LQR controller for the fast Nomoto model. If speed over ground (SOG) is chosen as the scheduling variable, a limit has to be set for when the switching between slow and fast controller should happen. When switching between the controllers, the control parameters change. This can generate a jump in the input of the USV, which should be avoided. Figure 2.9 illustrates the jump in the system's input.

To avoid this phenomena a bumpless transfer method should be implemented.

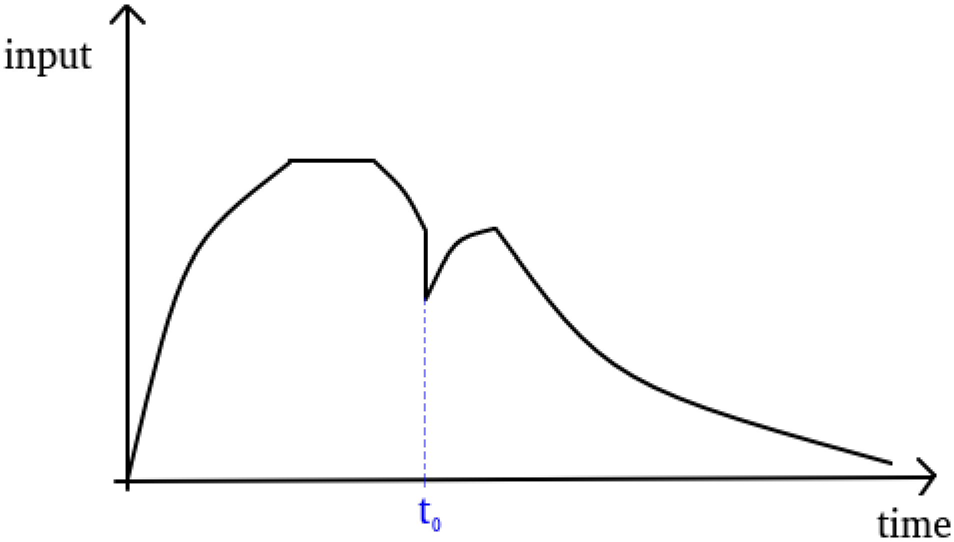


Figure 2.9: Bump transfer illustrated. At t_0 the gain scheduler switches to a new controller with other control parameters. This results in a jump in the input of the system.

For the Otter USV the input, δ, can be chosen as

$$\delta = (1-\alpha)\mathbf{K}_{\mathbf{slow}} + \alpha\mathbf{K}_{\mathbf{fast}} \qquad \alpha \in \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{2.39}$$

In Equation 2.39 α is a constant. However, by using α as a function which depends on the SOG it can be normalised and manipulated to a slower response by using mathematical functions. This is the purpose of α. Then alpha will interpolate between the two controllers.

If the function of α is sufficiently slow, this gives a bumpless transfer when the controllers are switching. The α depends on the scheduling variable, in this case the speed over ground, SOG. From Shamma and Athans (1990) two fundamental guidelines are presented as

1. The scheduling variable should vary slowly
2. The scheduling variable should capture the system's nonlinearities

The final input of the Otter USV with bumpless transfer is

$$\delta = (1-\alpha(U))\mathbf{K}_{\mathbf{slow}} + \alpha(U)\mathbf{K}_{\mathbf{fast}} \qquad \alpha(U) \in \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{2.40}$$

The reason SOG is chosen as the scheduling variable is because the heading dynamics depends on the USV's velocity, capturing the nonlinearities.

Introducing a function that depends on the scheduling variable, the input, δ, has a way of smoothly switching between the two controllers. The α function is divided into areas. Each area can have a function that is designed to schedule the controllers with the appropriate rate.

Chapter 3

Method

In this chapter the implementation of the Otter USV system will be described. The simulator is implemented in Matlab and Simulink. The first section will explain how the heading model of the Otter USV is implemented. The USV model is made by interpolating between two Nomoto models with respect to the surge velocity. Furthermore, to be able to use gain scheduling. A model of the scheduling variable, surge velocity, has to be made. This is done by using data from earlier acceleration tests and fit to a first order model.

The next section deals with the linear quadratic regulator. First, I explain the controllability and observability matrices calculated using the state space model with integral action Equation 3.4. In Section 3.2.2 the implementation of the control output where a precompensator is calculated for good tracking. Moreover, a description of how the linear controllers are tuned with respect to their corresponding model is presented.

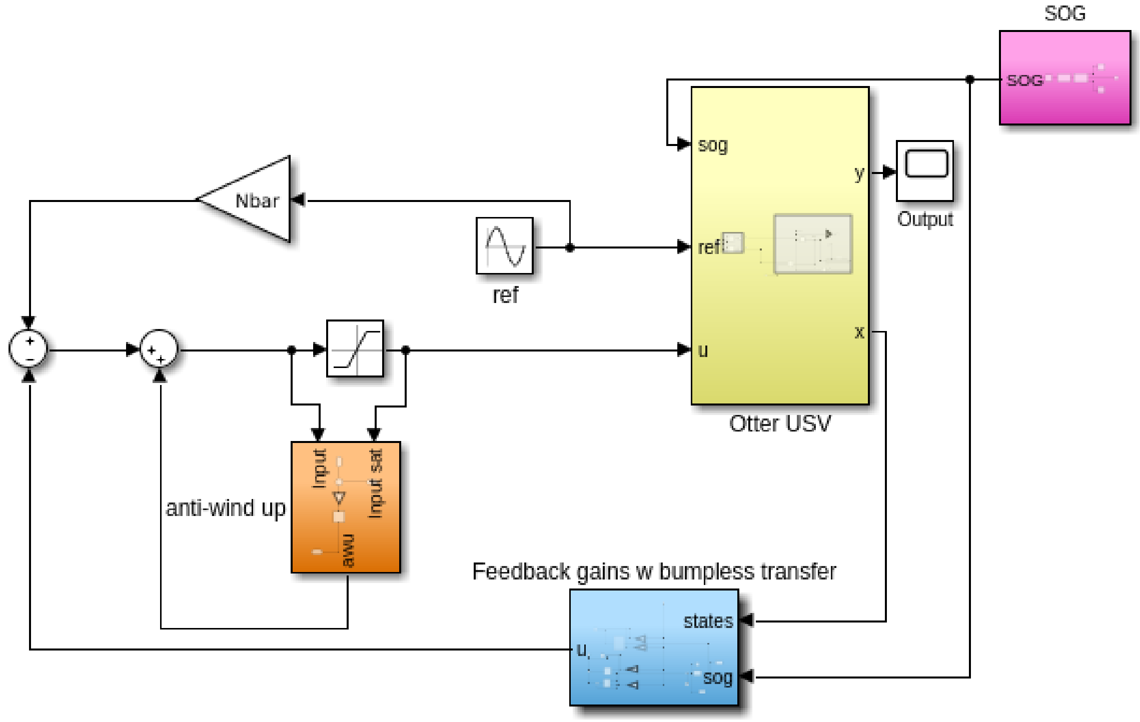


Figure 3.1: Overview of the whole system. Each subsystem has its own colour and will be explained in detail in this chapter.

3.1 Models of the Otter USV

This section will describe which models have been used to describe the heading dynamics and the surge velocity of the Otter USV.

3.1.1 Heading Dynamics

In the simulations for the heading dynamics for the Otter USV, the two Nomoto models proposed in Steindal (2018) are used.

$$\frac{r}{u_{1\,fast}} = \frac{0.0008}{0.5s+1} \tag{3.1}$$

$$\frac{r}{u_{1\,slow}} = \frac{0.0017}{Ts+1} \tag{3.2}$$

However, these two models were made for low velocities and high velocities, and in Steindal (2018) it is concluded that these two models are not particularly accurate. The report suggests that this is most likely because of the two propellers rotating the same direction when moving forward. The subsequent consequence is that the Otter USV has a higher turning rate to port side than to starboard side. A solution to this was to add two correction terms, which depend on the thrust from each propeller. The correction terms are not included in the simulated model because a mapping between the propellers RPM (rotations per minutes) and the velocity were unavailable when the simulator was made.

The Nomoto models are used as the simulated system because of a lack of better models for the Otter USV's heading dynamics. Furthermore, it is assumed that the Nomoto parameters are a fair approximation of the USV's heading dynamics. Based on this assumption the model parameters can be interpolated based on the measured speed over ground, SOG, where it is also assumed that SOG = surge velocity. The Otter USV's model parameters will change according to the vessel's speed over ground. The implementation in Matlab/Simulink is depicted in figure 3.2.

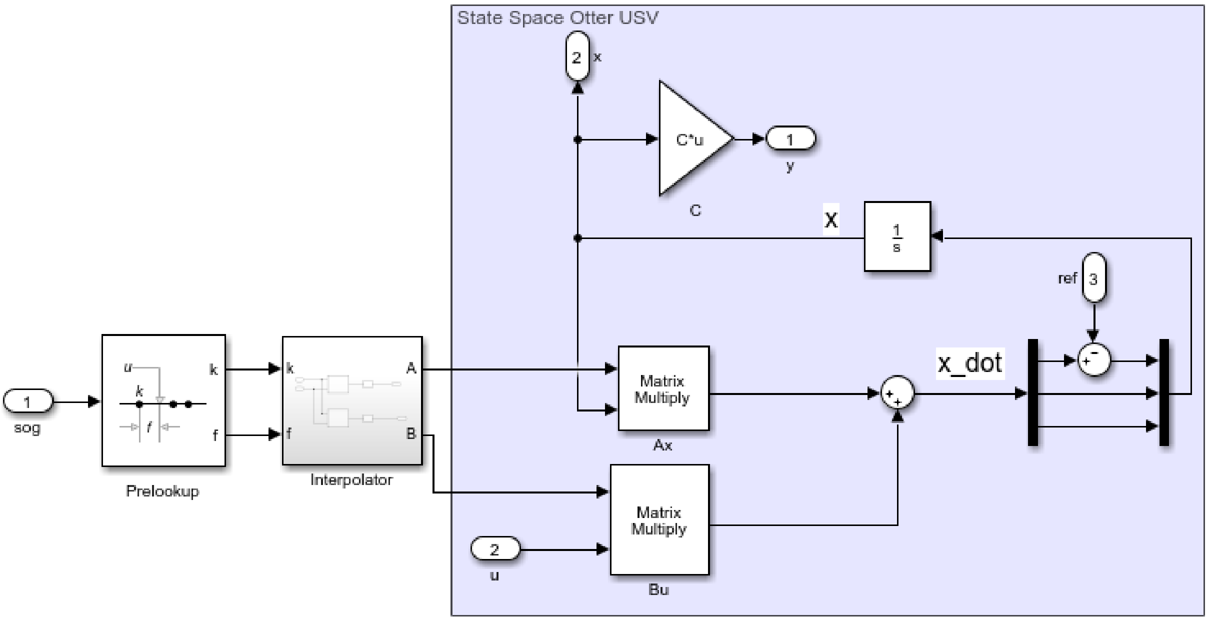


Figure 3.2: Implementation of the Otter USV heading dynamics in Simulink.

3.1.2 State-Space of The Otter USV and Adding Integral Action

In the Nomoto model the parameters K and T are assumed to be changing with the velocity of the vessel. There are two reasons for putting the system in a state space form. The first reason is that state space matrices are necessary to be able to solve the Riccati equation,2.29, and find the optimal gains for the linear quadratic regulator, LQR. The second reason is that it will be easier and more clear to interpolate the values of the parameters when they are matrix elements.

Equation 2.20 can be written in state space form, where the state vector $\mathbf{x} = \begin{bmatrix} \psi & r \end{bmatrix}^\top$. This gives

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T} \end{bmatrix} \begin{bmatrix} \psi \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K}{T} \end{bmatrix} u \tag{3.3a}$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \psi \\ r \end{bmatrix} \tag{3.3b}$$

Looking ahead for the LQR to be able to handle constant disturbances, an integral term is augmented to the state space system. The theory behind this is described in Section 2.2.1. For the Otter USV the state space system becomes

$$\begin{bmatrix} \dot{e} \\ \dot{\psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{T} \end{bmatrix} \begin{bmatrix} e \\ \psi \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{K}{T} \end{bmatrix} u \tag{3.4a}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e \\ \psi \\ r \end{bmatrix} \tag{3.4b}$$

where $\dot{e} = \psi - \psi_{ref}$, which gives the integration of the error in state $e = \int \psi - \psi_{ref} d\tau$.

3.1.3 Interpolation of the State Space Matrices

Looking at the augmented state space system, Equation 3.4, the time constant, T, is located in the $\mathbf{A}$ matrix as element $a_{3,3}$ and in matrix $\mathbf{B}$ element $b_{3,1}$ both Nomoto parameters are found.

The two Nomoto models for slow and fast turning depend on the vessel's velocity. Since there are only two models, the mid part of the velocity specter is missing. To account for this, the slow and fast Nomoto models are interpolated with respect to the velocity of the USV. Since the two models were made with the configuration of the Otter USV with a sonar, the maximum speed of the vessel is $2\,m/s$. Two additional sets of the system matrices, $\mathbf{A}$ and $\mathbf{B}$, are calculated using a linear function where the slope of the K-line and T-line is calculated based on the two Nomoto models from Steindal (2018) and the y-interception of the lines are the model parameters from the slow Nomoto model. The following Matlab code was used

```
% Specify the scheduling variable's range.
u = 0:2;                              % Speed over ground in m/s

% Specify K parameter
K = -0.0003*u + 0.0017;               % Assuming linear relationship

% Specify T parameter
T = -0.0667*u + 0.7;                  % Assuming linear relationship

% Compute linear system at a given SOG value.
for i = 1:length(u)
    A = [0 1 0;
         0 0 1;
         0 0 -1/T(i)];

    B = [0; 0; K(i)/T(i)];

    C = [1 0 0;
         0 1 0;
         0 0 1];

    sys(:,:,i) = ss(A,B,C,0);
end
```

The two additional sets of matrices are added to be used in the interpolator, which is depicted in Figure 3.3. The first block is called pre-lookup, which locates the position of the gain scheduling variable, speed over ground (SOG), on the set of intervals defined by the breakpoint data. The block outputs an index k and a fraction f, where k specifies the interval containing the input and f is the input's normalised position on the interval (Mathworks, 2019).

The next block is Simulink's matrix interpolator block, where linear interpolation is chosen for both matrices. The last block is a reshape block to change the $\mathbf{A}$ to a 3×3 matrix and $\mathbf{B}$ to a 3×1 matrix.

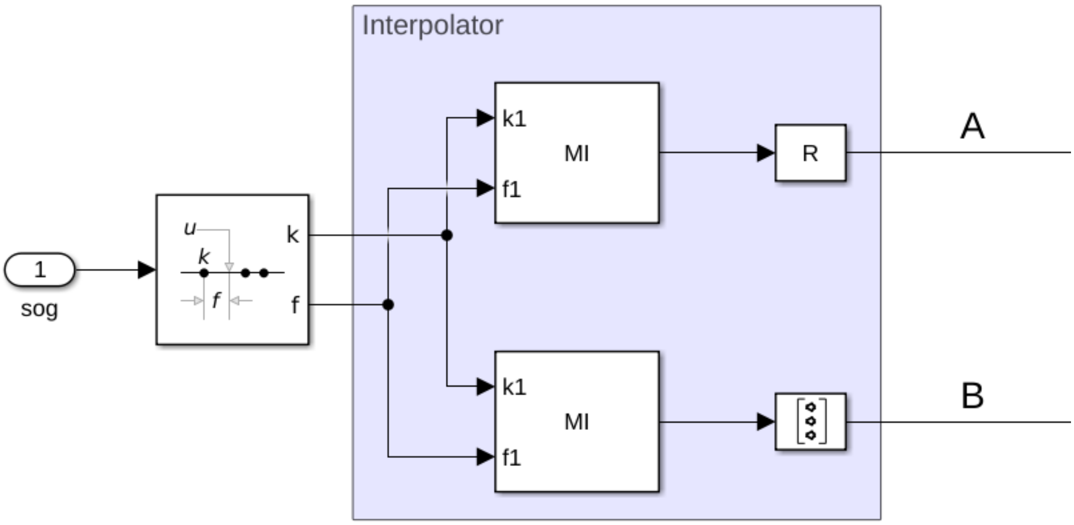


Figure 3.3: Matrix interpolator implemented in Simulink.

3.1.4 Model of the Scheduling Variable

Since the heading dynamics depend on the surge velocity of the vessel, it is natural to choose the vessel's forward velocity as the scheduling variable. The speed of the Otter USV is measured by a GPS as speed over ground, i.e. measured speed relative to the surface of the earth. An assumption that the Nomoto model builds on, is that the vessel's forward speed is slowly varying. Meaning that the vessel's speed is approximately the same as the vessel's surge velocity. This is derived in Section 2.1.3. Furthermore it is assumed that speed over ground measures the surge velocity of the USV. In order to make a reasonable simulator, a model of the vessel's surge dynamics was made.

Acceleration data was collected by employees at Maritime Robotics, in Fall 2018. The tests were executed so that the Otter USV accelerated from 0 to 100% of maximum throttle, with intervals of 10. These data were fitted to a first order linear model and the results are summarised in Table 3.1. Due to logging problems acceleration data for $0-20\%, 0-70\%, 0-80\%$ and $0-100\%$ were not saved.

To obtain stability of the system it is important that the dynamics of the scheduling variable is slower than the controlled heading dynamics. Based on the time constant, T, for the surge speed model, it can be concluded that surge dynamics are approximately 4 times slower than the heading dynamics.

The first model from 0-10% was excluded because the thrust was too low to overcome the resistance in the water. The final model for the surge velocity was obtained by taking the average of the time constant parameters (parameter located in the denominator) in

Table 3.1. The following surge model was obtained

$$U = \frac{1}{3.87s + 1} \tag{3.5}$$

Table 3.1: Results of acceleration tests for the Otter USV

0-10%	0-20%	0-30%	0-40%	0-50%
$\frac{0.304}{6.38s+1}$	Invalid test	$\frac{1.074}{3.71s+1}$	$\frac{1.508}{3.88s+1}$	$\frac{1.85}{3.4s+1}$
0-60%	0-70%	0-80%	0-90%	0-100%
$\frac{1.82}{4.83s+1}$	Invalid test	Invalid test	$\frac{2.738}{3.51s+1}$	Invalid test

3.2 Implementation of the Linear Quadratic Regulator

3.2.1 Controllability and Observability matrices

In Section 2.1.6, controllability is explained as a necessary property for the system while observability is an advantage, and may be necessary for estimating states. The first step before making the LQR controller is to check if the system is controllable and observable. This is done by checking the ranks of the controllability matrix and observability matrix. The controllability matrix becomes

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \frac{K}{T} \\ 0 & \frac{K}{T} & -\frac{K}{T^2} \\ \frac{K}{T} & -\frac{K}{T^2} & \frac{K}{T^3} \end{bmatrix} \tag{3.6}$$

which has full row rank equal to 3, which means the system is controllable.

The next step is to calculate the observability matrix

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{T} \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{T} \\ 0 & 0 & \frac{1}{T^2} \end{bmatrix} \tag{3.7}$$

which has full column rank equal to 3, which means the system is observable. However, since all states can be measured by sensors, it is not necessary that the system is observable.

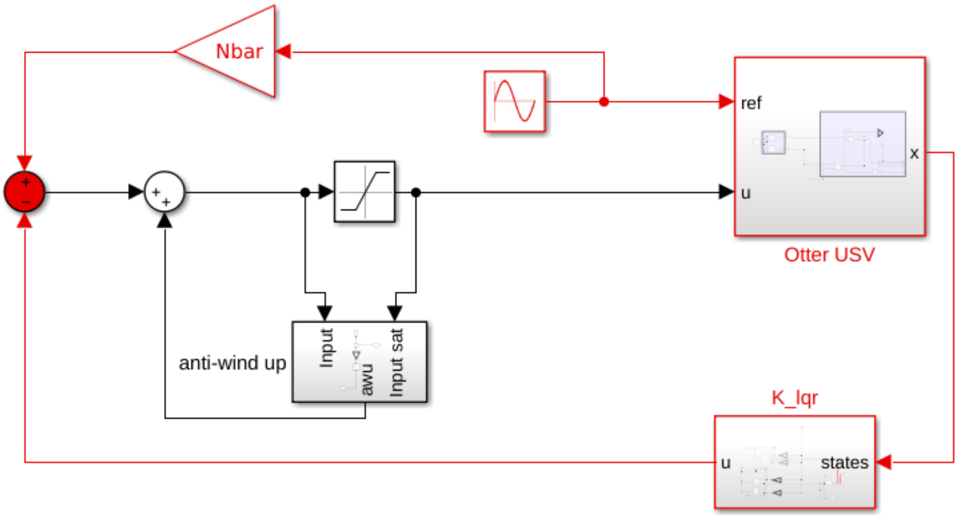


Figure 3.4: Simulink implementation of the control output.

3.2.2 Implementation of LQR with Tracking

The control output is defined as

$$u(t) = \bar{N}\psi_{ref} - \mathbf{K}_{lqr}\mathbf{x} \tag{3.8}$$

As depicted in Figure 3.4 full state feedback does not compare the reference with the measured output. It compares all the states, which are multiplied with the feedback matrix $\mathbf{K}_{lqr}$, to the reference. To compensate for this, a precompensator is calculated based on the system matrices. The formula and code is borrowed and adjusted from Messner et al. (2017). The matlab code used to calculate $\bar{N}$ is

```
T = 0.63;
K = 0.0014;

A = [0 1; 0 -1/T];
B = [0;K/T];
C = [1 0];
D = 0;

LQR_slow = [624.5 2682.3 1569.3];
LQR_fast = [830.7 3032.2 1268.5];

LQR_avg = [(LQR_slow(2)+LQR_fast(2))/2 (LQR_slow(3)+LQR_fast(3))/2];

s = size(A,1);
Z = [zeros([1,s]) 1];
N = inv([A,B;C,D])*Z';
Nx = N(1:s);
Nu = N(1+s);
Nbar = Nu + LQR_avg*Nx;
```

Notice that $\bar{N}$ is equal to the first element in state feedback matrix, LQR_{avg}. Furthermore, $\bar{N}$ is calculated based on the linear model of the Otter USV and in addition the precompensator is outside of the feedback loop. Thus, if the model is not perfect or there exist unknown disturbances, the precompensator will not be able to correct them and there will be a steady-state error. Moreover, since integral control is added, it can be used to eliminate steady-state error, even if model uncertainty and step disturbances are present.

A disadvantage of integral control is that the error must first grow before it can be corrected for. Hence, it might take a little time before the system responds. The precompensator can anticipate the steady-state offset since it is calculated based on the model of the system. A convenient approach is to combine the precompensator with integral control to obtain the advantages of each approach (Messner et al., 2017).

3.2.3 Tuning of LQR

Two LQR controllers are tuned based on their corresponding linear model. The model parameters for the two Nomoto models are summarised in Table 3.2. The tuning was done by designing a simulink model where the system is the corresponding fast/slow model and then tune the controller to a satisfactory response. This means that there is a 100% match between the model of the system and the model the controller is based on. This is not realistic, because the Otter USV is a non-linear system.

An overview of the controller gains and model paramters for the fast and slow controller is given in Table 3.2.

The tuning matrices for the heading controllers are chosen such that error in yaw angle is expensive. The Q matrix is chosen large and R is chosen low. This is to get a fast system response and at the same time be able to use lots of controller output to be able to reach the desired angle. By increasing the constant ($= 300$) the system response will be faster. However, the input can get oscillations, which will cause wear and tear on the thrusters. This should be avoided, hence the relatively low constant.

The integral state is penalised harder in the controller for fast velocities because the USV is using more time to achieve the desired heading angle. The yaw state is penalised the hardest because the control objective is to obtain close tracking of the heading angle, so it is desired that it should be expensive to have error in yaw.

The yaw rate is penalised to even out the controller output. The controller for slow velocities needs a high value to prevent the input, $u(t)$, to overshoot a little before it settles. This is also the case for the controller for fast velocities, yet the effect is not that prominent for this controller.

$$\mathbf{Q}_{slow} = \begin{bmatrix} 130 & 0 & 0 \\ 0 & 1500 & 0 \\ 0 & 0 & 700 \end{bmatrix} \cdot 300 \quad R = 0.1 \tag{3.9}$$

$$\mathbf{Q}_{fast} = \begin{bmatrix} 230 & 0 & 0 \\ 0 & 1670 & 0 \\ 0 & 0 & 330 \end{bmatrix} \cdot 300 \quad R = 0.1 \tag{3.10}$$

The controller gains have the following property

$$\mathbf{K}_{slow/fast} = [\textit{Integral gain} \quad \textit{Proportional gain} \quad \textit{Derivative gain}] \tag{3.11}$$

Table 3.2: Summary of the controller gains and model parameters

	Model parameters		Controller gains
Slow	$K = 0.0017$,	$T = 0.7$	$\mathbf{K}_{slow} = [624.5, 2682.3, 1569.3]$
Fast	$K = 0.0008$,	$T = 0.5$	$\mathbf{K}_{fast} = [830.7, 3032.2, 1268.5]$

3.2.4 Bumpless Transfer

Two controllers are now tuned for the slow and fast Nomoto models. The next step is to implement the bumpless transfer technique described in Section 2.4. The essence is to make a function, $\alpha(U)$, that interpolates between the two controllers.

The function operates in the interval between 0 and 1, and depends on the velocity of the Otter USV. This means that when $\alpha(U)$ is low, the slow controller dominates while when $\alpha(U)$ increases, the fast controller dominates. The implementation of the bumpless transfer method in Simulink is depicted in Figure 3.5.

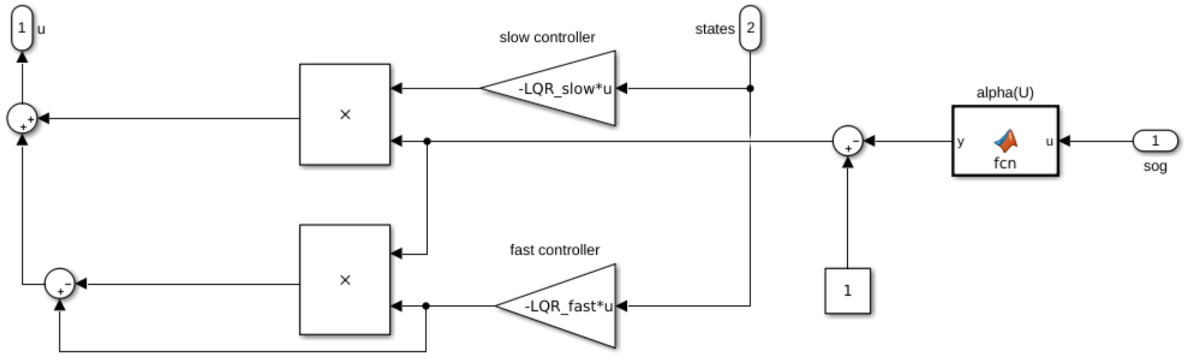


Figure 3.5: Simulink implementation of the controllers with bumpless transfer.

The $\alpha(U)$-function is implemented using a Matlab function block with the following code

```
function y = fcn(U)
   % Max speed of otter with sonar is 2 m/s
    max_speed = 2;
    U_norm = min(U/max_speed,1);         % Normalizing the speed

    fast_ex = 4;                         % Tuneable exponent. Small values give steeper slope

    fast_b = (0.5-0.5^fast_ex)/(1-0.5^fast_ex);


    if U_norm >= 0 && U_norm <= .3
        alpha = min(U_norm,0.3);
    elseif U_norm > .3 && U_norm < .5
        alpha = min(U_norm,0.5);
    else
        alpha = min((1-fast_b)*U_norm^fast_ex + fast_b,1);
    end

y = alpha;
```

From 0 to 0.5 $\alpha(U)$ is a linear function. From 0.5 the curve is changed to obtain a slower change in $\alpha(U)$, this is due to quicker acceleration in the velocity and to make sure that $\alpha(U)$ does not change too fast, which can cause instability. $\alpha(U)$ is depicted in Figure 3.6.

The exponent is set equal to 4 to give a slow change when switching to the fast controller. Using an interpolator between two controllers is an easy solution to obtain bumpless transfer.

However, if more controllers were to be implemented, gain scheduling using PID controllers should be easier to implement and keep track of. PID controllers have the advantage of having an easy structure and the control parameters can be accessed independently, in contrast with the gain matrix, $\mathbf{K}_{LQR}$, which is calculated using the Riccati equation. In addition, there exist multiple bumpless transfer methods for PID controllers.

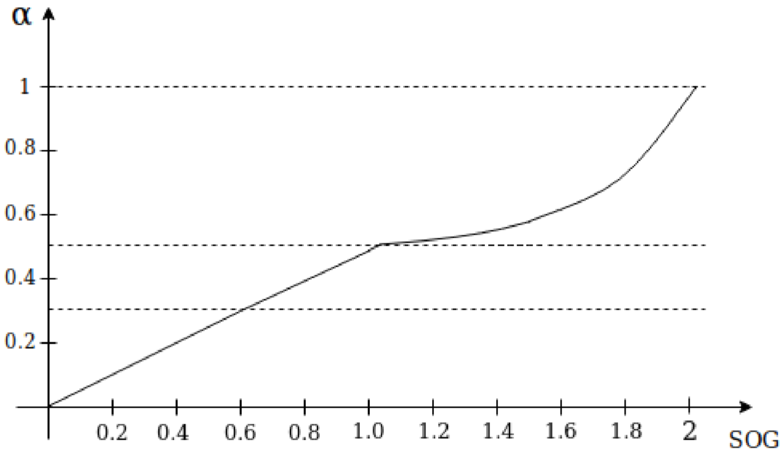


Figure 3.6: α as a function of the gain scheduling variable.

3.3 Actuator Limits

The Otter USV is controlled by two electrical thrusters. The thrusters have a limitation in angular acceleration and a maximum angular velocity. Furthermore, the input to the Nomoto model is torque, which is given by the differences in thrust from the actuators. For the Otter USV, torque is defined as (Steindal, 2018)

$$u_1 = Q_p = (F_{port} - F_{stb})a \quad [\text{Nm}] \tag{3.12}$$

where F is the thrust from each thruster and a is the moment arm.

Hence, for the Nomoto model the saturation limits are the maximum amount of torque the propellers can produce. The highest turning rate is achieved when one propeller has maximum forward velocity and the other propeller has maximum reverse velocity. However, since the propellers are not symmetric, a thruster in reverse is less effective. The thrust curve for the Otter USV was acquired by performing a bollard pull test as described in Steindal (2018). Based on the measurements from the bollard pull test the thrust curve was estimated to the following thruster model

$$F = k \cdot \omega|\omega| \tag{3.13}$$

where F is the thrust, k is a the thrust parameter and ω is the angular velocity of the thruster.

The measured points were found with the following matlab code

```
%% torque_limits.m
clear all
close all
clc
% Measurements of the thrusters are in rpm, need to convert it to rad/s
% Convert rpm to rad/s
rpm2rads = 2*pi/60;

%% Measured pull
% Forward pull
Fp = [0 0.4 1.3 3.0 5.4 8.8 12.4 17.3 22.7 24.4 24.0]'*9.81;   % [N]
thruster_f = [0 121.4 244.6 358.7 490.5 623.2 726.1 844.6 968.0 972.0 970.7]'*rpm2rads;

% Backward pull
Bp = -[0 0 0.7 1.9 3.5 5.4 7.4 10.8 13.4 13.5 13.6]'*9.81;    % [N]
thruster_b = -[0 122.0 244.4 359.5 493.4 615.4 728.1 857.1 964.4 970.9 968.1]'*rpm2rads;

%% Setup for using fit()
s = fitoptions('Method','NonlinearLeastSquares',...
               'Lower',min(Bp),...
               'Upper',max(Fp),...
               'Startpoint',0);

f = fittype('a*x^n','problem','n','options',s);

[nlsq_B,gof] = fit(thruster_b,Bp,f,'problem',2);

[nlsq_F,gof2] = fit(thruster_f,Fp,f,'problem',2);
% nlsq_x finds the thrust parameter in rad/s (B: 0.01289   F: 0.02141)

figure(4)
y1 = 0.01289*abs(thruster_b).*thruster_b;
hold on
plot(thruster_b,y1)
scatter(thruster_b, Bp)

y2 = 0.02141*abs(thruster_f).*thruster_f;
plot(thruster_f,y2)
scatter(thruster_f, Fp)

xlabel("\omega [rad/s]")
ylabel("Thrust [N]")
title("Thrust curve for Otter USV")

hold off
```

Based on the measured force the maximum torque for the Otter USV can be calculated by using Equation 3.12. The maximum amount of torque is given if one thruster is rotating with maximum forward speed and the other rotates with maximum reverse speed. This is calculated by using the following matlab code:

```
clear all
close all
clc

%% Moment arm for one thruster
a  = 0.395;                                                    %[m]

%% Total measured force for both thrusters FORWARD
Fp = [0 0.4 1.3 3.0 5.4 8.8 12.4 17.3 22.7 24.4 24.0]'*9.81;   %[N]

%% Force from 1 thruster FORWARD
Fp_half = Fp/2;                                                %[N]

%% Total measured force for both thrusters REVERSE
Bp = -[0 0 0.7 1.9 3.5 5.4 7.4 10.8 13.4 13.5 13.6]'*9.81;     %[N]

%% Force from 1 thruster REVERSE
Bp_half = Bp/2;                                                %[N]

%% Maximum force
F1 = max(Fp_half)                                             %[N]
F2 = min(Bp_half)                                             %[N]

Tp = (F1-F2)*a                                                %[Nm]
```

3.3.1 Integral Windup

The effect of the saturation in torque is the possibility of integral windup. The integral windup as a phenomenon is described in Section 2.2.3. When the Otter USV is tracking a larger heading angle, the controller will compute a large controller output, which is too large and it will be saturated. The integrator will continue integrating the error until it starts to decrease. However, the integrator term has to decrease first, which results in a large settling time. The solution is to implement one of the anti windup techniques described in Section 2.2.3. The logic anti windup technique, Equation 2.37, is the easiest one if the integral gain is easy to access. However, since the gains are located in the $\mathbf{K}_{slow/fast}$ matrix, it is easier to implement the back calculation anti windup technique, Equation 2.38.

The gain K_{aw} is tuned to be 0.4 which gave the fastest response and no overshoot. Smaller values gave large overshoots, and larger values gave slower response.

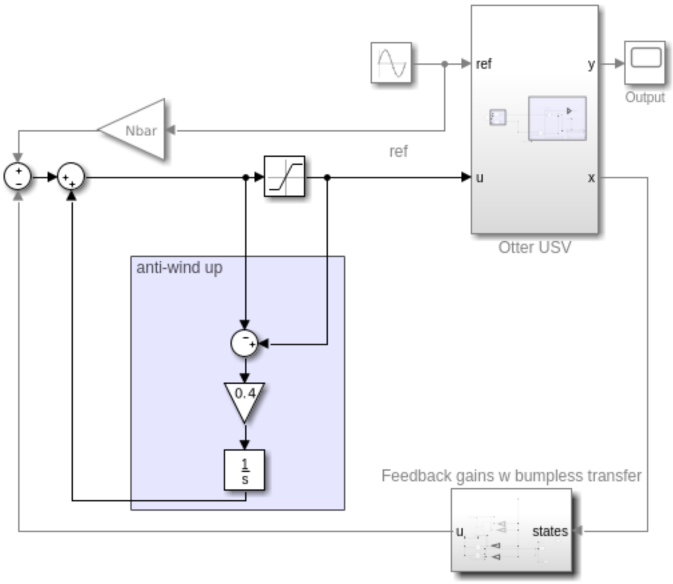


Figure 3.7: Implementation of anti-wind up.

Chapter 4

Results

4.1 Thrust Curve for the Otter USV

The measured forces from the bollard pull test are depicted in Figure 4.1 as circles, and the thrust parameters are summarised in Table 4.1. The thrust curve in Figure 4.1 shows that the propellers are less effective when rotating backwards. However, the thruster model described by Equation 3.13 is a good fit in both directions. Comparing the maximum measured pull, it can be seen that the propellers are approximately 55% less effective when rotating backwards. This is because the propellers are designed for better performance when rotating in the positive direction.

Table 4.1: Results of the estimation of the Otter USV thrust curve

	Thrust parameter, k	Maximum pull
Forward pull (0-80%)	$k = 0.02141$	22.7 kg
Forward pull (0-100%)	$k = 0.02216$	24.4 kg
Reverse pull	$k = 0.01289$	13.6 kg

There are two data points that differ from the rest, which are highlighted as green circles in Figure 4.1. The two data points are measurements for 90 and 100% of maximum throttle. The measurements are a little off, so two thruster curves were estimated with and without these two measurements.

In Steindal (2018) a bollard pull test was performed only for forward pull. The result of that test was $k = 0.02164$, which is close to the estimated result for the forward pull from 0 to 80%.

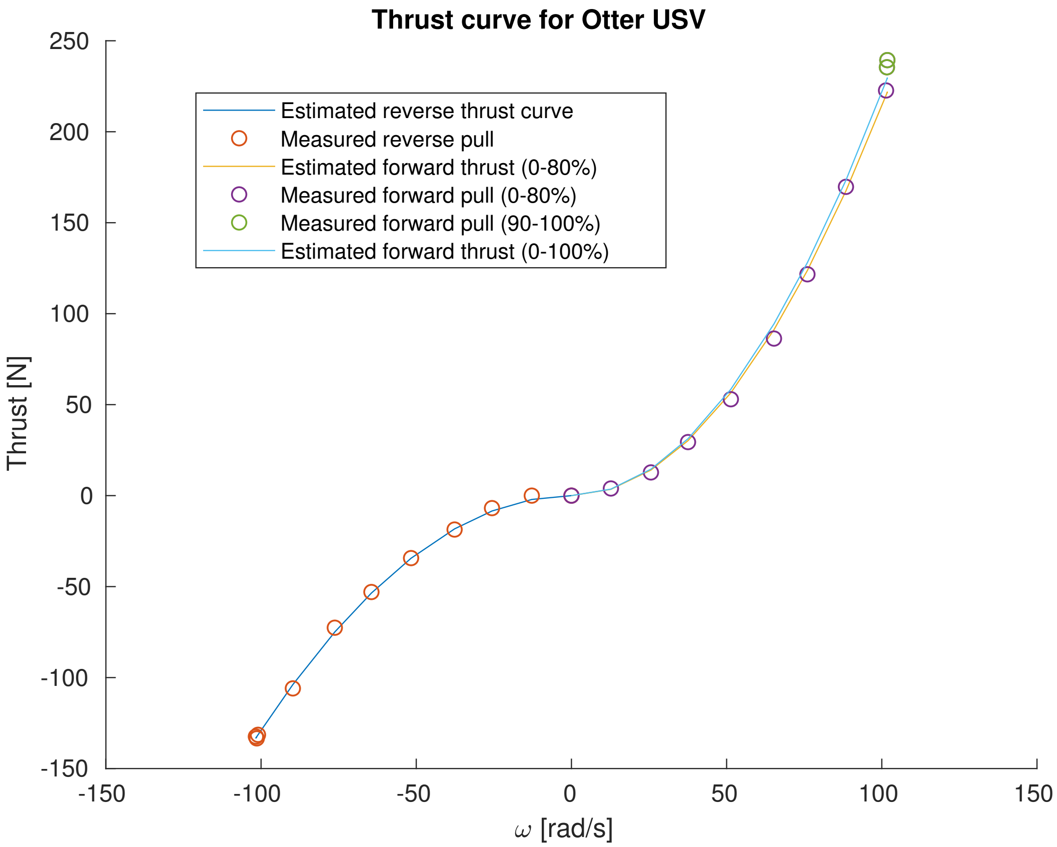


Figure 4.1: Thrust curve for the Otter USV. Notice that the measured thrust is the sum of both thrusters. The solid lines are the estimated thrust curve, where the thrust parameters are summarised in Table 4.1.

The saturation limits are calculated using the maximum measured pull to calculate the maximum torque to port and starboard. The maximum measured pull in forward direction is $24.4\ kg$ and $13.6\ kg$ in reverse pull. These two values are used in the script `torque_limits.m` where the maximum torque is calculated to be

$$u(t) = \begin{cases} u_{max} = 73.6, & u > u_{max} \\ u(t), & u_{min} \leq u \leq u_{max} \\ u_{min} = -73.6, & u < u_{min} \end{cases} \tag{4.1}$$

4.2 Tuning of LQR

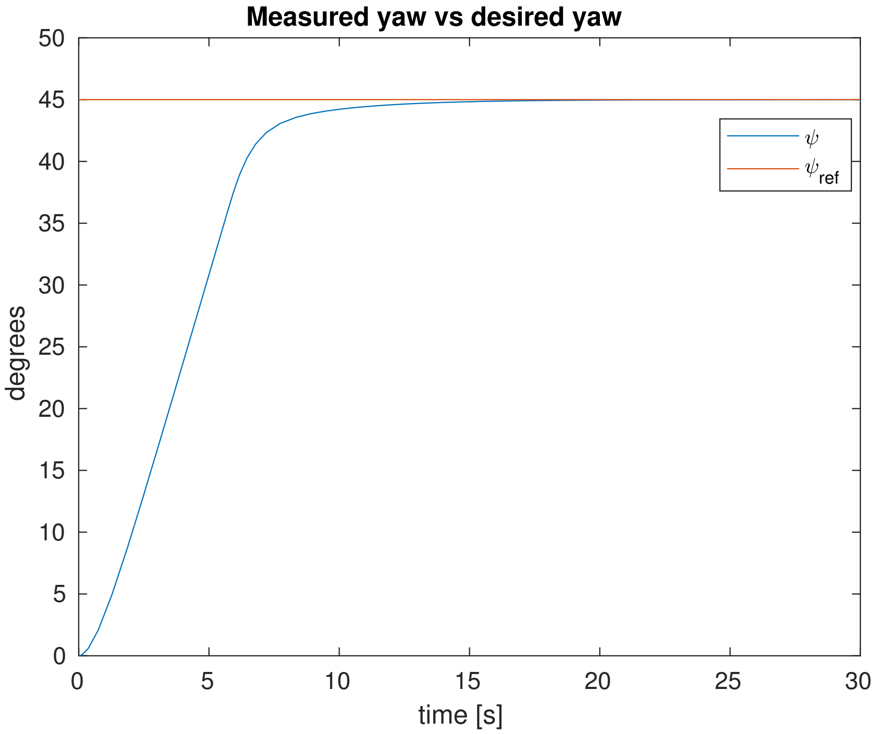
4.2.1 Nomoto Model for Slow Velocities

The LQR controller response for the Nomoto model for slow velocities is depicted in Figure 4.2a. The Otter USV reaches 90% of the reference after approximately 6 seconds, the last $5°$ are reached 6 seconds later. The advantage of the slower response at the end is that it avoids overshooting. Since the Otter USV has a catamaran hull, the settling time will be long, if the Otter USV overshoots.

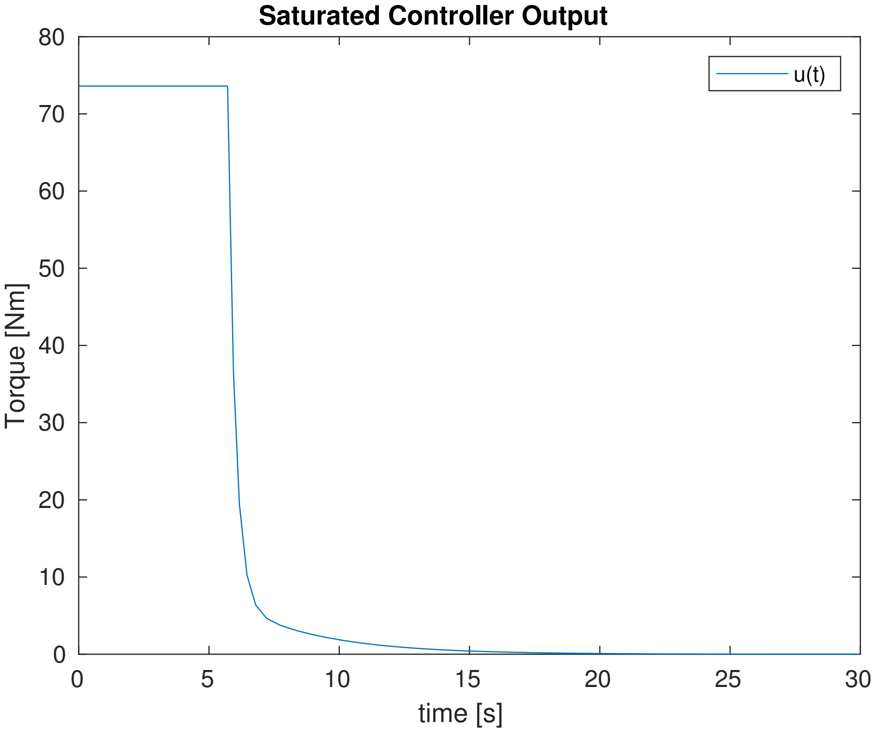
The controller output is depicted in 4.2b. The controller output is maxed out and saturated for the first 6 seconds, then it sharply drops towards 4 Nm and slowly goes towards 0. This corresponds well with the step response of the Otter USV, where the last $5°$ uses the same amount of time as the first 90% of the reference.

4.2.2 Nomoto Model for Fast Velocities

The response of the Nomoto model for fast velocities is depicted in Figure 4.3a. The response is slower, reaching 90% after approximately 12 seconds, however, the input decays fast and evenly such that the last $5°$are reached 2 seconds later. In total, the model for fast velocities is 2 seconds slower than the Nomoto model for slow velocities.

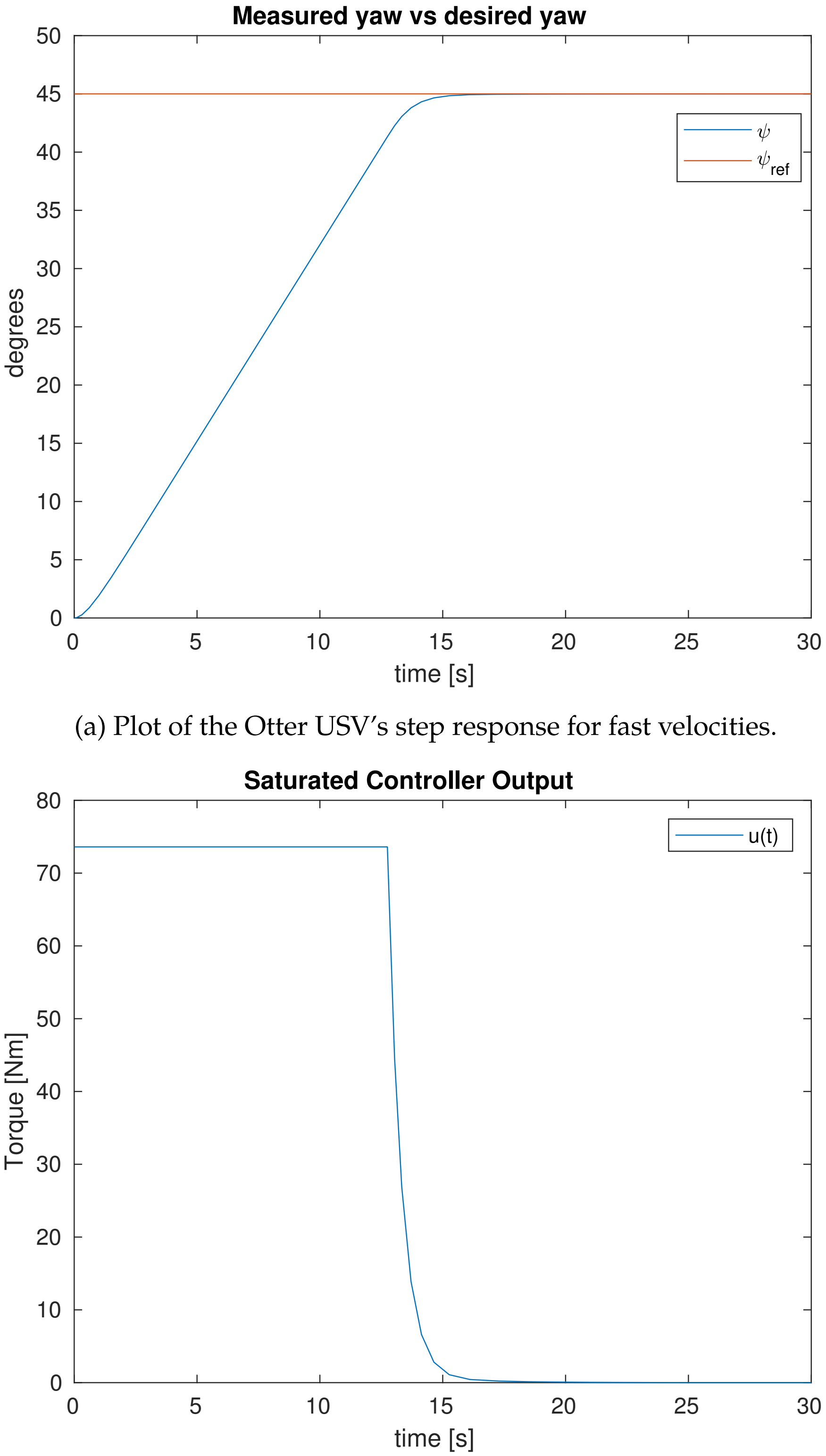


(a) Plot of the Otter USV's step response for slow velocities.



(b) The control output for slow velocities.

Figure 4.2: Results of the tuning for the Nomoto model for slow velocities.



(a) Plot of the Otter USV's step response for fast velocities.

(b) The control output for fast velocities.

Figure 4.3: Results of the tuning for the Nomoto model for fast velocities.

4.3 The Effect of Bumpless Transfer

In this section we compare the controllers with and without bumpless transfer, in order to show that the bumpless transfer is indeed important compared to a gain scheduled controller without a bumpless transfer. For the controller without bumpless transfer, a switch will decide which controller is active, based on the velocity.

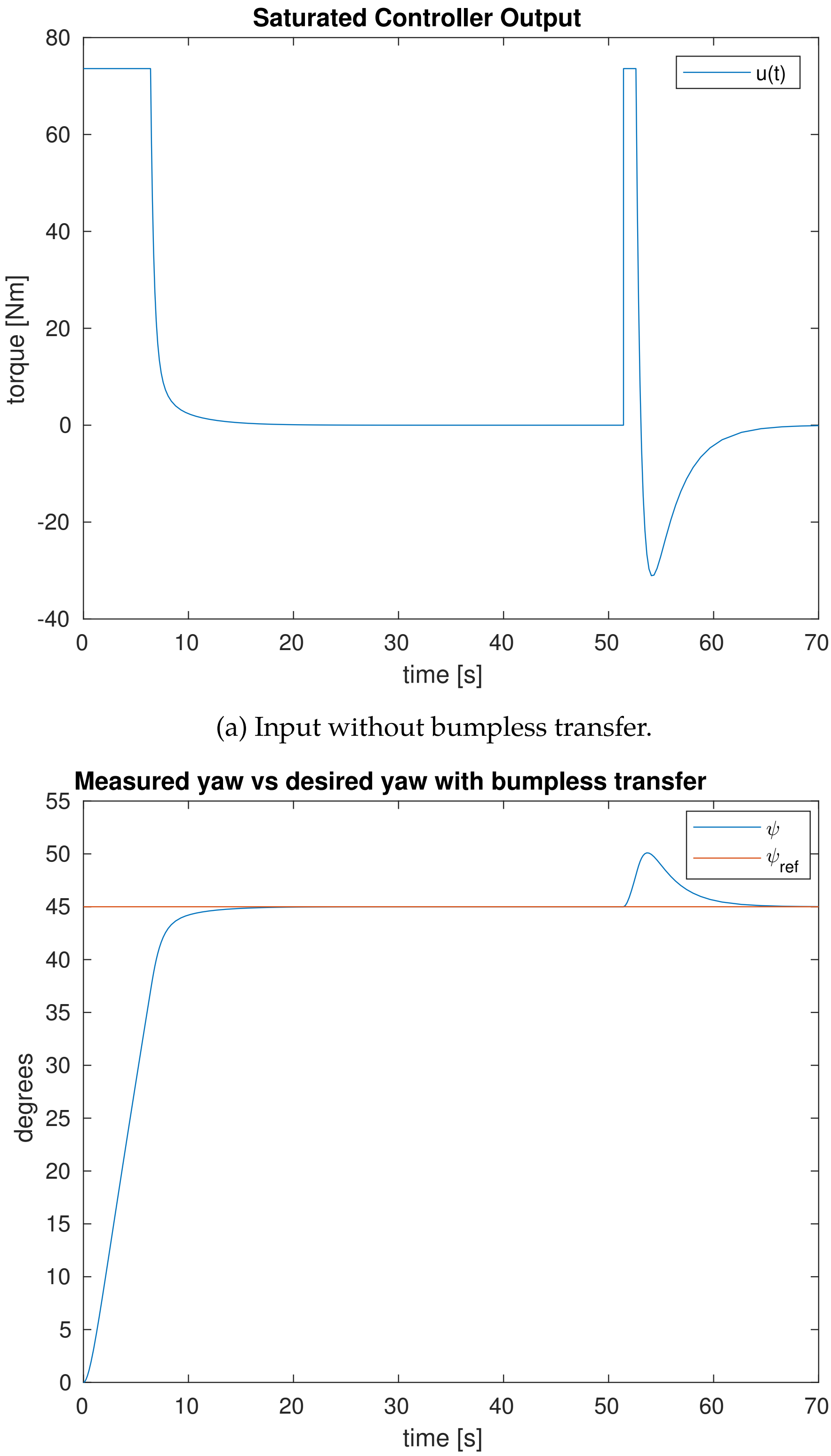
When the surge velocity hits $1\,m/s$, the controller is switched, depending on if the acceleration is positive or negative. The surge velocity is depicted in Figure 4.5 and is the same for both simulations, with and without bumpless transfer.

The effect of the switching is prominent in Figure 4.4a, where a large bump happens after 50 seconds. The large and quick jump in the input is physically impossible for the thrusters to fulfil, because they need time to adjust.

In the heading response, we can observe an overshoot after 50 seconds. If the error in yaw had been exactly 0 this would not occur. However, the error is not exactly 0 and when the state feedback matrix is multiplied with the states, the error increases with approximately $5°$.

When we compare simulations without bumpless transfer to simulations with bumpless transfer, there are considerable improvements in the controller output, Figure 4.6a, where there are no sudden jumps in the input due to the function $\alpha(U)$. Figure 4.6c shows that the values of α is slowed down after the velocity has reached 50% of maximum. This is to suppress sudden jumps when the velocity is high and thus try to slow down the transfer between the two controllers.

The bump in the heading response is decreased with $3°$, which is depicted in Figure 4.6b. However, since these bumps are very small, meaning that the error is very small due to rounding in Simulink, this can be ignored and should not be a problem in real life where it will be rounded to 0.



(a) Input without bumpless transfer.

(b) The heading angle response without bumpless transfer.

Figure 4.4: Simulation of the Otter USV without bumpless transfer.

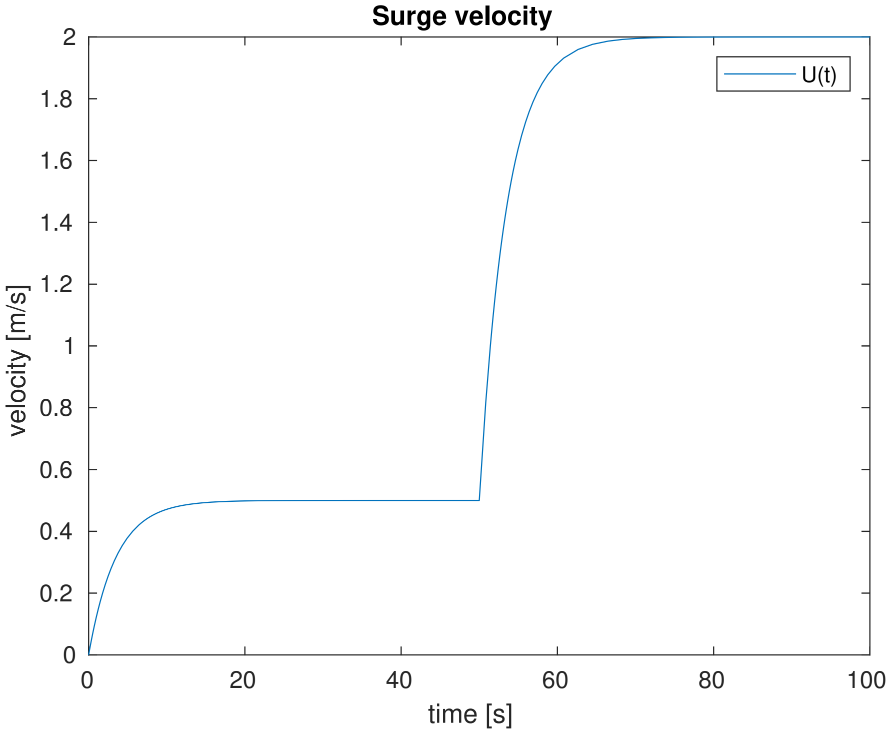
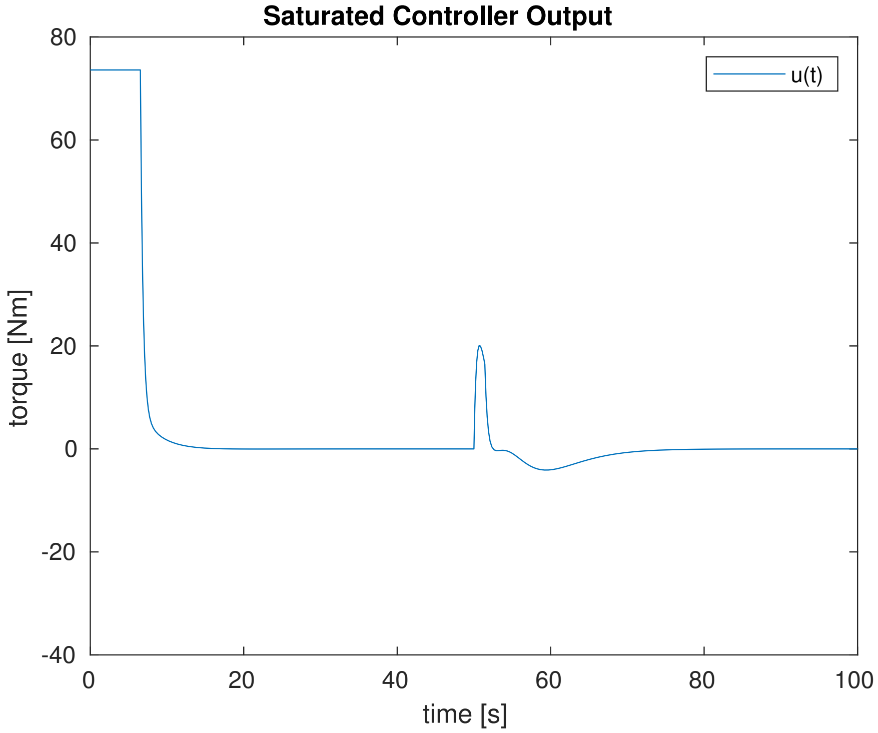
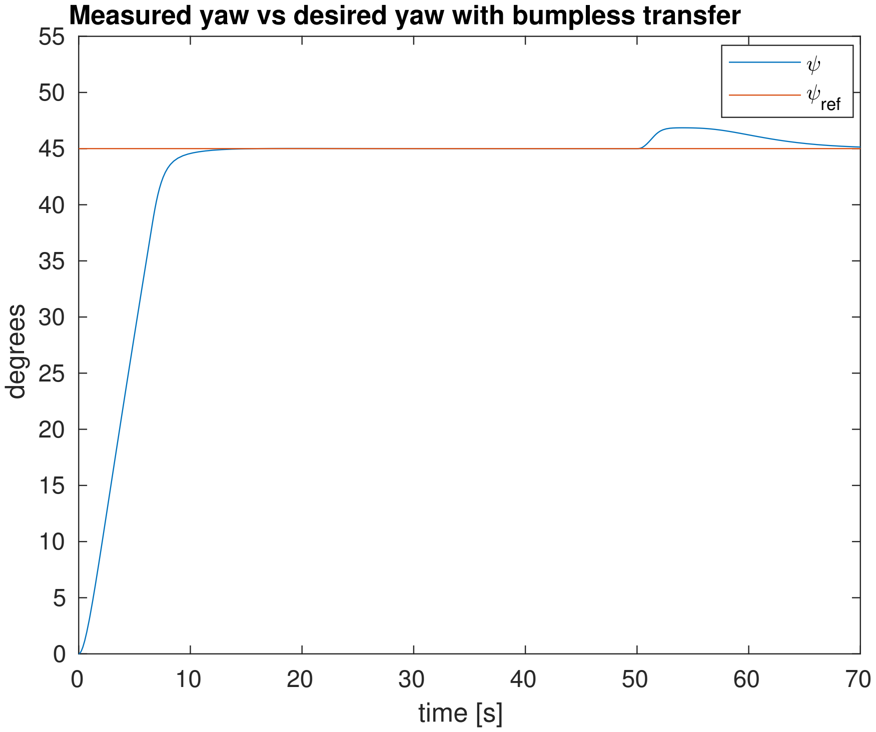


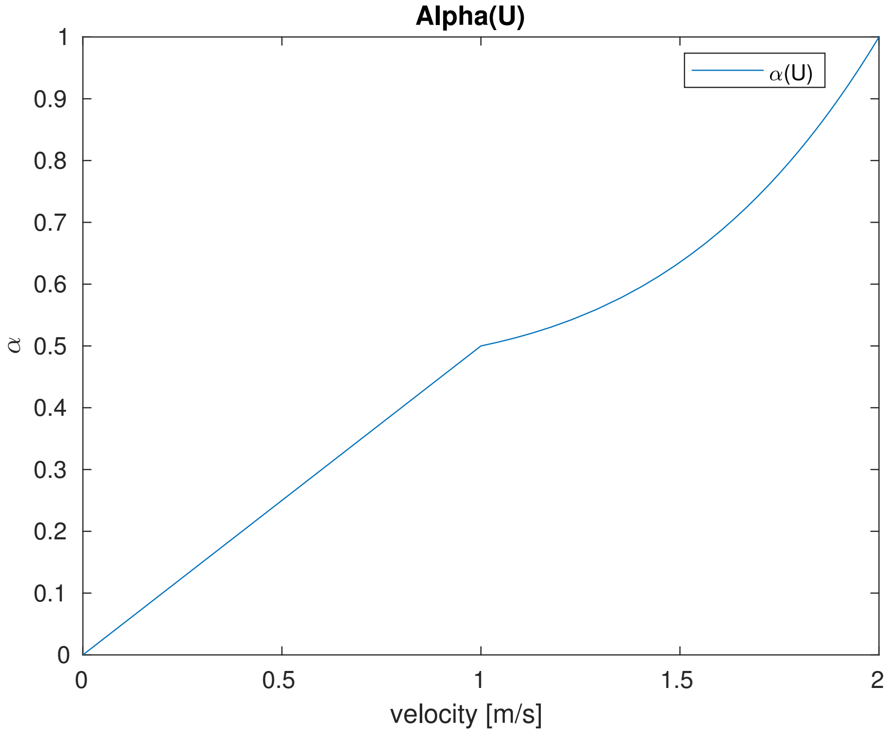
Figure 4.5: Surge velocity



(a) Input with bumpless transfer.



(b) The heading angle response with bumpless transfer.



(c) $\alpha(U)$, the controller interpolator function

Figure 4.6: Simulation of the Otter USV with bumpless transfer.

4.4 Gain scheduled LQR with Bumpless Transfer Versus Static LQR

Step Response

In this section, the results of a step response of the proposed gain scheduled controller with bumpless transfer will be compared to a controller with constant state feedback, where $\alpha(U) = 0.5$. The proposed bumpless transfer technique, Equation 2.39, uses $\alpha(U)$ as a interpolator between the two gain scheduled controllers. The value of α is depicted in Figure 4.8c, where it can be seen that the proposed gain scheduled controller varies with the velocity. The other controller has $\alpha = 0.5$, which gives a controller performing in the middle of the two gain scheduled controllers.

In Figure 4.8a the gain scheduled controller overshoots with $1°$, while the controller with constant α has no overshoot and reaches the reference faster. The controller outputs, Figure 4.8b, are almost identical, but it can be seen that the controller with constant α uses less input, which prevents it from overshooting.

The surge velocity in this simulation is depicted in Figure 4.8d, it goes from 0 to $2\,m/s$.

The main purpose of this is to check if the controller with constant α is managing to control the system, even though the model parameters are interpolated with respect to the velocity. Interestingly, as Figure 4.8a shows, the controller is managing the change in model parameters and performs a little better than the gain scheduled controller.

Tracking a Time Varying Reference

Since the step response of the two controllers do not manage to distinguish the two controllers, a more advanced simulation is presented in Figure 4.9. The surge velocity is chosen to vary from low to high and back to low to check how the controllers manage the transitions. Comparing Figure 4.9a and 4.9b the first zoomed window of the plots show that the two controllers are performing quite similar. The second zoomed window shows the transition from high to low speed and it can be observed that the gain scheduled controller has a small bump before it regulates back to the reference, while the constant controller has a very tight tracking the whole time.

Saturated Input

The input for the tracking case is depicted in Figure 4.7. The interesting parts of this plot is the transition times for the velocity. It can be seen that the gain scheduled controller with bumpless transfer struggles to obtain a smooth input when the velocity changes. This plot also indicates that the tuning of the two controllers are a bit aggressive. The bumps would be smaller if the controller parameters were smaller and closer, however this would slow down the system.

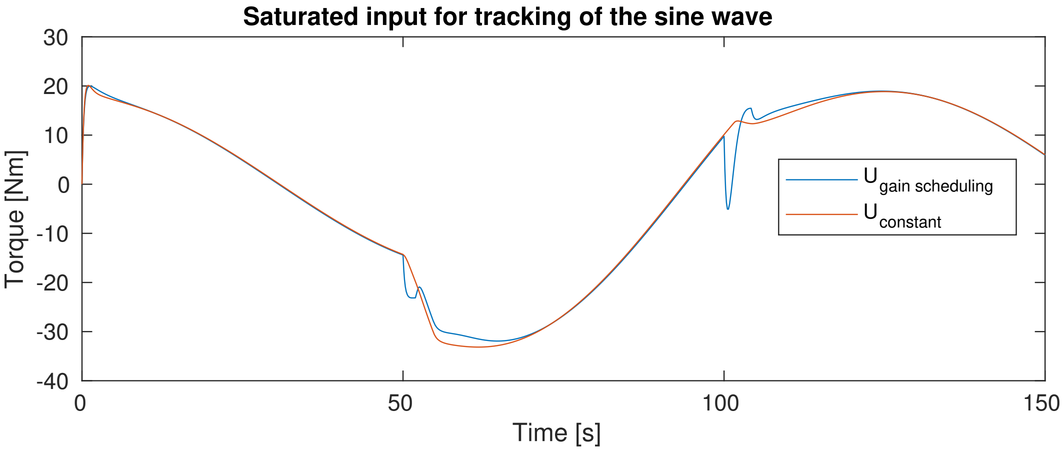
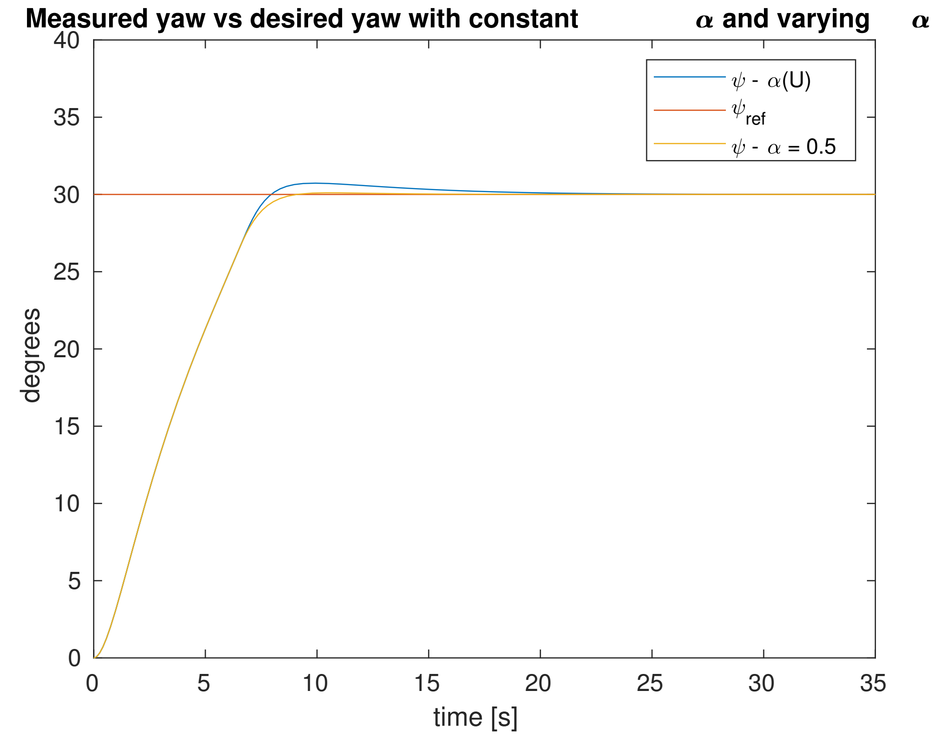
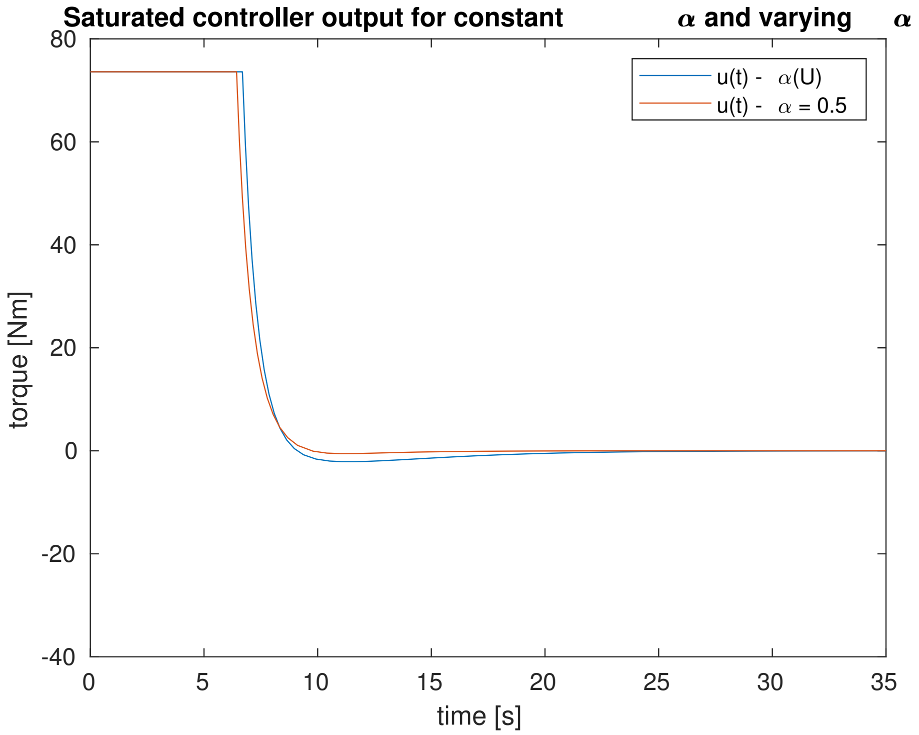


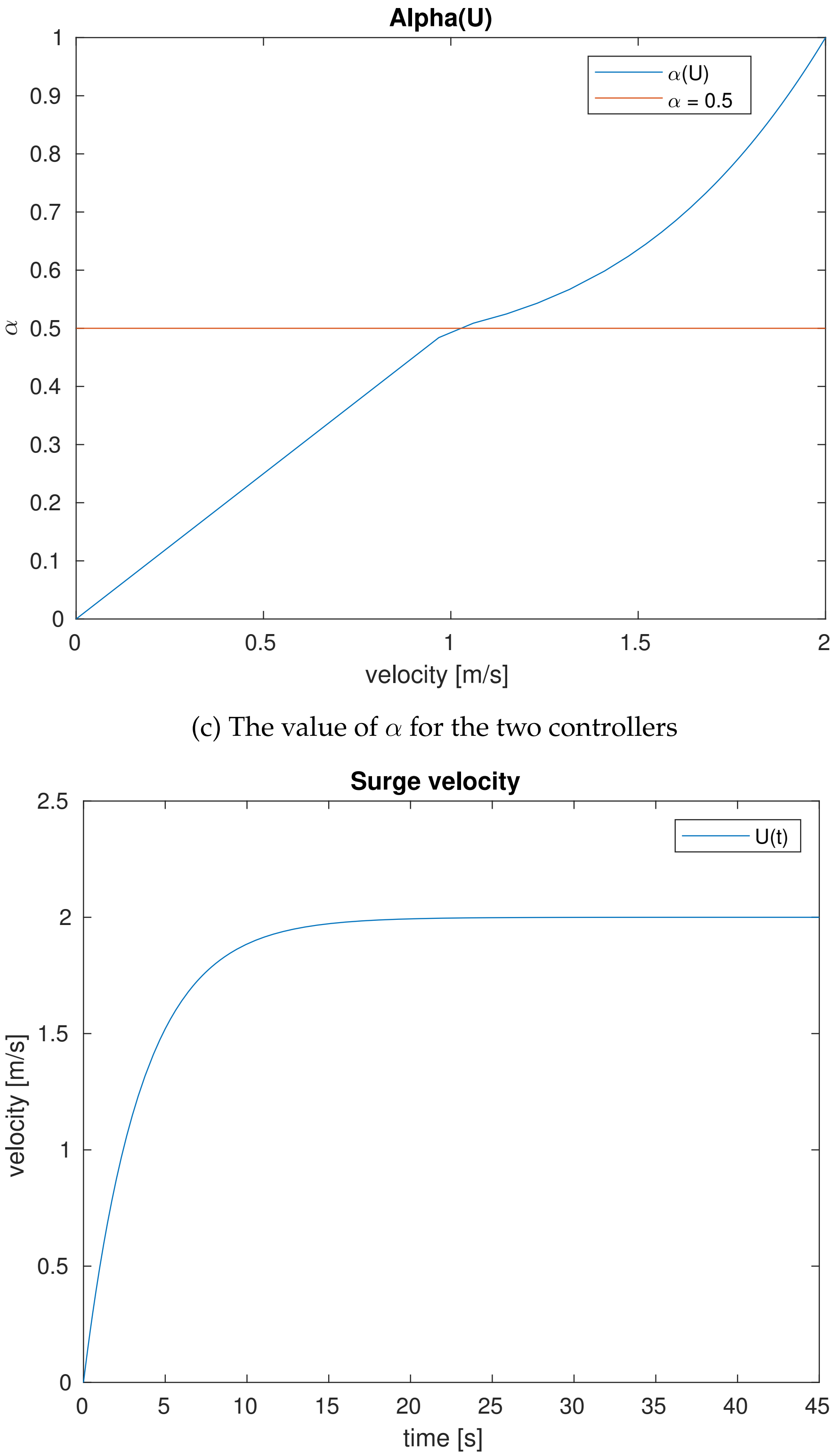
Figure 4.7: Saturated input of the two controllers



(a) Comparison of the heading angle with the different controllers



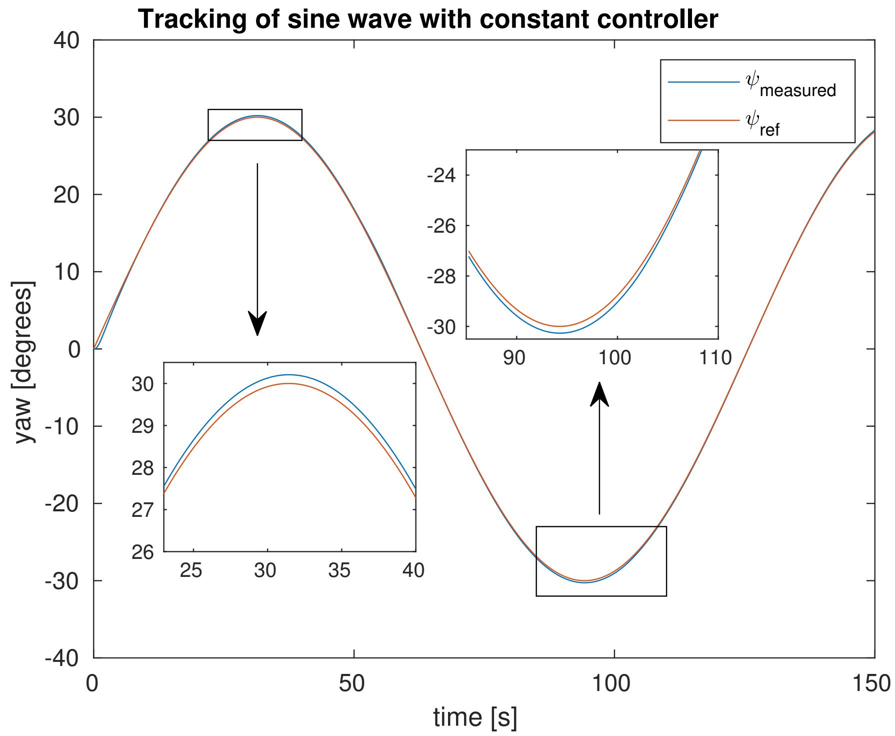
(b) Comparison of the controller output for the gain scheduled controller and the constant LQR



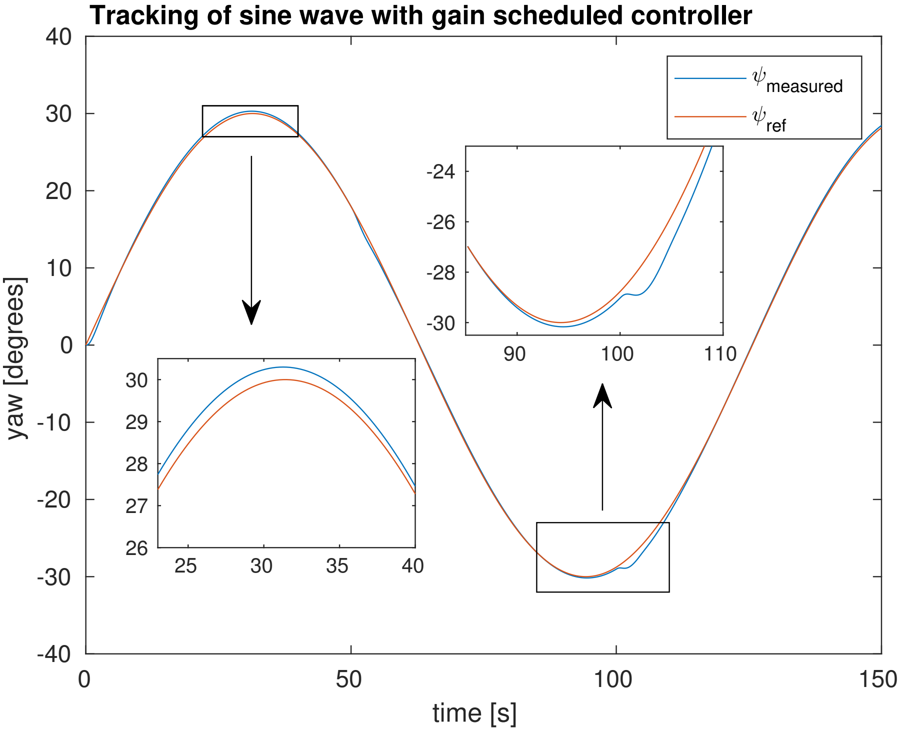
(c) The value of α for the two controllers

(d) Surge velocity for the simulations

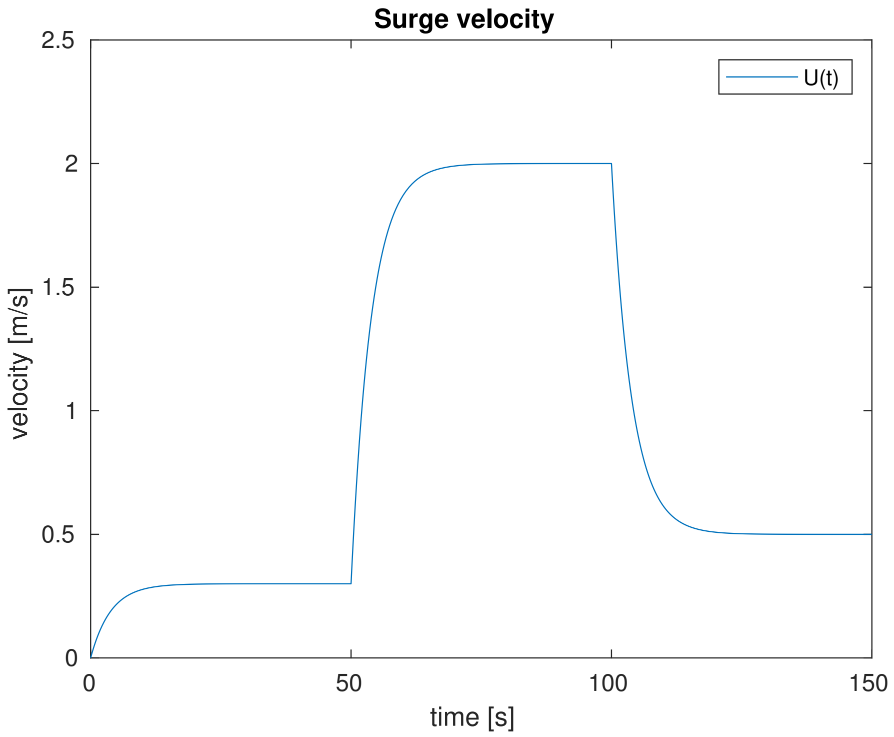
Figure 4.8: Simulation of the Otter USV given a step where the gain scheduled controller with bumpless transfer is compared with a controller with constant state feedback matrix



(a) Tracking of a sine wave with constant LQR controller



(b) Tracking of a sine wave with gain scheduled controller with bumpless transfer



(c) The surge velocity for the tracking case

Figure 4.9: Simulation of the Otter USV with time varying reference where the gain scheduled controller with bumpless transfer is compared with a controller with constant state feedback matrix

4.5 Discussion

4.5.1 Thrust Curve

The results of the thrust curve was that the curve for backwards rotation is not as power full as when the propellers are moving in forward direction. This means that the thrusters are not symmetrical and are less effective when spinning backwards, which is a regular property for propellers. Furthermore, this property is important to consider when with turning the Otter USV, especially if the Otter should use little space when turning.

The proper way to do this is to perform an opposing turn where one thruster is rotating forward and the other is rotating backwards. If it is assumed that the thrusters performs with equal thrust when rotating forward and backwards, the result for the Otter USV is that it will drift, which will increase the error assuming that the Otter is supposed to track a certain path. The controller output will then increase and the Otter USV will have an aggressive response when turning. If the turning is done by scaling the maximum forward thrust with a factor of 0.6, then the two thrusters will produce the same amount of thrust in opposite direction and presumably rotate around the center of mass and have a smooth turn.

The Otter USV was equipped with different pontoons for the bollard pull test performed in Steindal (2018), which can explain the different results. Moreover, the results differ with rather small variations which are expected anyway since it is difficult to reproduce exact experimental results even if there were no different pontoons.

4.5.2 Bumpless Transfer

The bumpless transfer technique used in this master thesis interpolates between the controller parameters. By using a function that takes values between 0 and 1 and depends on the scheduling variable it is possible to avoid jumps in the input. Studying the response in Figure 4.6 it can be seen that even though there are no sudden jumps in the input, there is a quite fast peak reaching approximately 20 Nm. This is the disadvantage of the interpolation method because it depends on the velocity, which is changing quite fast and the $\alpha(U)$ does not manage to slow down the changes in the control parameters very well. An alternative to this can be to make a controller with back-calculation. The method uses an integrator to reach the new control input and avoid a jump in the input. The advantage with this method is that a time constant can easily be used to tune the rapid change in the control input. This is a technique similar to the anti-windup with back-calculation (Åström, 2002).

4.5.3 Gain Scheduled Controller vs. Constant Control Parameters

An important property of the simulated system is that the model of the Otter USV used in the simulations is linear, which is the same as the two LQ controllers are based on. This is not realistic since the Otter USV is a non-linear system. To compensate for the lack of a better realistic model the model parameters were interpolated with respect to the surge velocity.

Another weakness and uncertainty is the model of the surge velocity, which was based on the average of only six acceleration tests. The main purpose of obtaining a model of the surge velocity was to have a realistic idea of how fast the scheduling variable changed. The obtained model of the surge velocity gave an indicator that the dynamics of the scheduling variable was slow enough compared to the Nomoto models, such that it could not make the system unstable when used as the scheduling variable.

As the results in Figure 4.8 showed, the step response is a bit quicker for the gain scheduled controller with bumpless transfer. However, it overshoots with 1° and uses a while to settle. The overshoot is very small and can be ignored, but the response is just a second faster than the controller with constant α, so the result for the gain scheduled controller is not superior compared to a single LQR.

The results of tracking a time varying reference showed more clearly that the constant controller is in fact better than the gain scheduled controller with bumpless transfer. This is particularly depicted in the input Figure 4.7, where the the gain scheduled controller has problems to avoid small bumps in the input. This also indicates that the bumpless transfer technique is not always managing to do its job.

A possible explanation for why the static controller with $\alpha = 0.5$ is better than the gain scheduled controller can be viewed in Figure 4.10. The Nomoto model is a linearisation of the yaw rate, which depends on the USV's velocity. Furthermore, by having two linearised Nomoto models that are made for different velocities, a more accurate description of the yaw rate can be obtained. The red and green dotted lines represent the two Nomoto models for low and high velocities. The pink line represents the model where $\alpha = 0.5$, and it can be seen that the pink line is also a fair approximation of the nonlinear system. This can explain why the static controller performs as good as the gain scheduled controller.

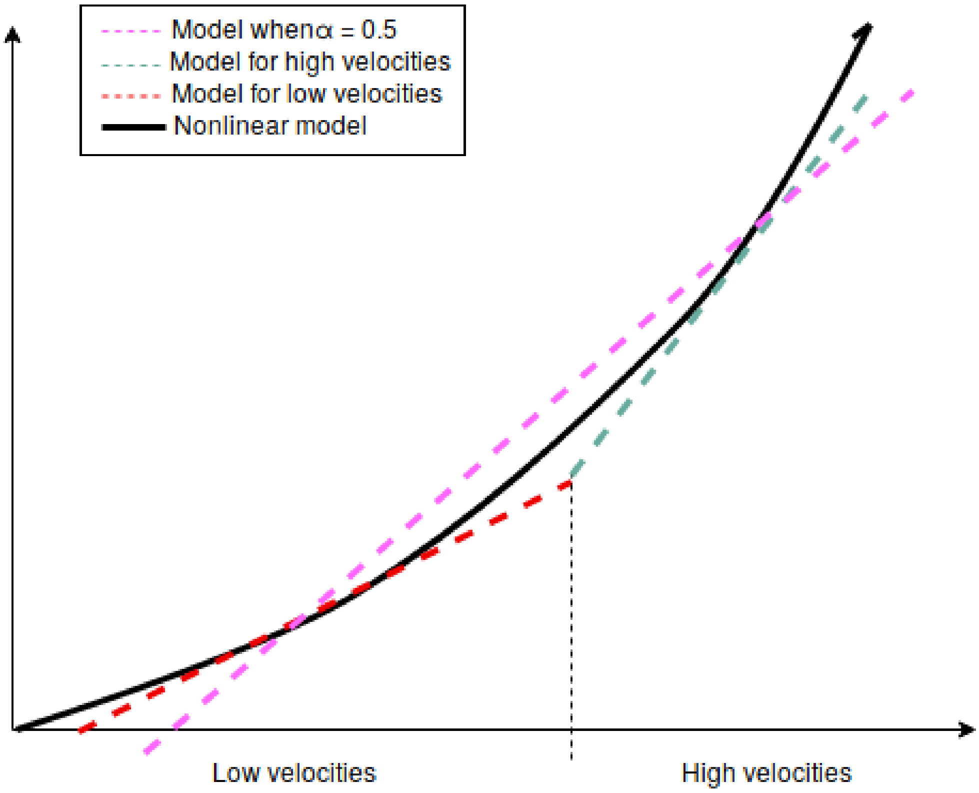


Figure 4.10: Theoretic figure of the nonlinear heading system

Chapter 5

Conclusion

The overall goal in this master thesis was to make a gain scheduled controller with bumpless transfer and check if this is a suitable control scheme for the Otter USV. A model of the USV's heading dynamics was made by interpolating between the model parameters of two Nomoto models. The interpolation was done with respect to the Otter USV's surge velocity. Furthermore, a gain scheduled controller was made using the Nomoto models to obtain two LQ controllers that can be switched between based on the vessel's velocity. To avoid sudden jumps in the plant input the control parameters were interpolated using a function $\alpha(U)$ which also depends on the velocity.

The gain scheduled controller does not perform better than a static LQ controller. Based on the results it can be concluded that there are no advantages of using a gain scheduled controller for the Otter USV. Furthermore, it should be noted that it is difficult to be certain of this conclusion due to 100% match between the mathematical model of the Otter and the model the controller is based on.

5.1 Recommendations for Further Work

The major problems encountered when making the controller is the model it is based on. If a nonlinear model of the Otter USV can be made it would be a considerable improvement. Moreover, the Nomoto models could also be improved by performing zigzag tests to get more information about the dynamics of the system. The gain scheduled controller and the static LQ controller could be implemented on the Otter USV and test performance, however, this should be done with caution because the control parameters can be too aggressive. Another possibility is to implement the gain scheduled controller with the bumpless transfer technique called "bumpless transfer with back-calculation".

Bibliography

Arpaia, P., Ballarino, A., Daponte, V., Montenero, G., and Svelto, C. (2014). Smart monitoring system based on adaptive current control for superconducting cable test. *Review of Scientific Instruments*.

Athans, M. and Falb, P. L. (1966). *Optimal Control*. McGraw-Hill Book Company, New York, NY.

Bendtsen, J. D., Stoustrup, J., and Trangbaek, K. (2005). Bumpless transfer between observer-based gain scheduled controllers. *International Journal of Control*.

Breivik, M. (2010). *Topics in Guided Motion Control of Marine Vehicles*. PhD thesis, Norwegian University of Science and Technology, Faculty of Information Technology, Mathematics and Electrical Engineering Department of Engineering Cybernetics.

Chen, C. T. (1999). *Linear System Theory And Design*. Oxford University Press.

Fossen, T. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons.

Frazzoli, E. (2010). Feedback control systems 16.30 fall 2010. https://ocw.mit.edu/. Massachusetts Institute of Technology: MIT OpenCouseWare. Accessed: 2019-06-04.

Khalil, H. (2001). *Nonlinear Systems*. Pearson.

Kim, S.-H. (2017). *Electric Motor Control: DC, AC, and BLDC Motors*. Elsevier Science.

Leith, D. J. and Leithead, W. E. (2000). Survey of gain-scheduling analysis and design.

Liu, Z., Zhang, Y., Yu, X., and Yuan, C. (2016). Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*.

Maritime Robotics (2019). the otter - the mobile usv system from maritime robotics. https://maritimerobotics.com/mariner-usv/otter/. Accessed: 2019-06-04.

Mathworks (2019). Prelookup - compute index and fraction for interpolation using prelookup block. https://se.mathworks.com/help/simulink/slref/prelookup.html. Accessed: 2019-05-31.

Messner, B., Tilbury, D., Hill, R., and Taylor, J. (2017). Control tutorials for matlab & simulink. http://ctms.engin.umich.edu/CTMS/. Collaboration of Carnegie Mellon University, University of Michigan and University of Detroit Mercy. Accessed: 2019-06-05.

Nomoto, K. (1960). Analysis of kempf's starndard maneuver test and proposed steering quality indices. *First Symposium on Ship Maneuverability*, Report 1461:275–304.

Nomoto, K., Taguchi, T., Honda, K., and Hirano, S. (1957). On the steering qualities of ships. *International Shipbuilding Progress*, 4:354–370.

Shamma, J. S. and Athans, M. (1990). Analysis of gain scheduled control for nonlinear plants. *IEEE Transactions on Automatic Control*.

SNAME (1950). The society of naval architects and marine engineers. nomenclature for treating the motion of a submerged body through a fluid. *Technical and Research Bulletin No. 1-5*.

Steindal, J. A. F. (2018). System identification of an unmanned surface vehicle with fixed dual thrusters. Project report in TTK4550 specialization project.

Triantafyllou, M. S. and Hover, F. S. (2003). 2.154 maneuvering and control of marine vehicles (13.49). fall 2004. https://ocw.mit.edu/. Massachusetts Institute of Technology: MIT OpenCouseWare. Accessed: 2019-07-01.

Åström, K. and Rundqwist, L. (1989). Integrator windup and how to avoid it. *American Control Conference, Pittsburgh, USA*.

Åström, K. J. (2002). Control system design. University of California, Santa Barbra, CA.