Hover and Forward Flight of an Autonomous UAV

- using Optimal Linear Control and Gain Scheduling

Master thesis June 2007

Group 1035b

AALBORG UNIVERSITY · AAU
Department of Control Engineering
Institute of Electronic Systems

AALBORG UNIVERSITY

Department of Control Engineering
Fredrik Bajers Vej 7C
DK-9220 Aalborg Ø
Phone.: +45 9635 8600
Web: http://www.control.aau.dk

TITLE:

Hover and forward flight of an Autonomous UAV using optimal linear control and gain scheduling

THEME:

Intelligent Autonomous Systems

PROJECT PERIOD:

9th-10th semester
September 2nd 2006 - June 7th 2007

PROJECT GROUP: 1035b

GROUP MEMBERS:

Teis Bæk
Mads Hammelsvang
Thomas Bæk Jørgensen

SUPERVISOR:

Anders la Cour-Harbo

NUMBER PRINTED: 5

NUMBER OF PAGES: 135

FINISHED: June 7th 2007

ABSTRACT:

This master's thesis concerns the design of linear control strategies for making an AUV capable of autonomous flight. The goal is to be able to complete level 1 in the International Aerial Robotics Competition, which involves flight over a distance of 3 km, as fast as possible.

The starting point for this thesis is taken in the developed nonlinear model of a reconfigured Bergen Industrial Twin helicopter, where the linear controllers are designed based on linearization of the nonlinear model.

Two different linear control design methods are used for the controller design; classic SISO control and optimal LQR control. Furthermore, a gain scheduling approach to perform scheduling between two controllers is introduced. As a high level controller a supervisory controller is designed to handle position control.

Hover control was designed using the two different linear control design methods. The classic SISO controllers was implemented as lead or lag compensators. Furthermore, a forward flight LQR controller was developed to be able to perform fast forward flight. The gain scheduling approach was implemented as an observer-based gain scheduler framework around the LQR hover and forward flight controllers.

The LQR hover and forward flight controllers, along with the observer-based gain scheduling controller, were all tested separately in three different level 1 specifications to be able to compare them. It was concluded that the observer-based gain scheduling approach was the most time efficient control method of the three.

PREFACE

This master's thesis has been written by group 1035b on the specialization of Intelligent Autonomous Systems at the Department of Control Engineering, Aalborg University, in the period between September 2nd 2006 and June 7th June 2007.

The thesis has been done as a part of the development of an autonomous helicopter based UAV at the Department of Control Engineering at Aalborg University for future participation in the IARC.

The thesis consists of three parts ”Helicopter model introduction and control system analysis”, ”Control system design” and ”Control system test and conclusion”, which contain several chapters each starting with a short introduction. Last in the thesis appendices are placed with supplementary subjects and are denoted with capital letters starting with A. Enclosed at the back of this thesis a CD is placed containing MATLAB© code, SIMULINK models, C code and a pdf copy of the thesis. For visualization in 3D the program GSIM is used (see Appendix A).

The thesis is intended for supervisors, examiner, control students and others that might have interest in linear control and gain scheduling on an autonomous helicopter.

Aalborg University, 2007

Teis Bæk

Mads Hammelsvang

Thomas Bæk Jørgensen

Reading instructions

References to literature are done by the Harvard method, where needed specific pages are added, e.g. [Sørensen, 1992, p.45]. Figures, equations, and tables are numbered consecutively within each chapter. References to equations are in addition made in parenthesis.

To clarify the difference between vectors and matrices these are written with bold lower-case letters and with bold capitals respectively. Variables and symbols stems, wherever possible, from the used literature.

Contents

1
INTRODUCTION

In several years the subject of autonomous flight with helicopters has been researched by different project groups and a single Ph.d. student at the Department of Control Engineering at Aalborg University. Two of the student projects done most recently are master's thesis motivated by the International Aerial Robotics Competition (IARC) 2006, organized by the Association for Unmanned Vehicle Systems International (AUVSI [1972]). In this competition an Unmanned Aerial Vehicle (UAV) is required autonomously to complete a number of levels involving different challenges inspired by different mission examples.

Inspired by the previous work done on autonomous flight and the IARC, this master's thesis focus is on designing linear control for an UAV making it capable of autonomous flight. For this purpose more advanced control methods are looked into in order to achieve this goal. Here the control method *gain scheduling* is chosen because of it ability to extend the range at which linear control of a given system can be performed.

In order to better understand the motivation behind the IARC, and thereby the basis for making an UAV performing autonomous flight, a short mission example involving a nuclear disaster is given.

One of four units at a nuclear reactor has exploded, and there are no survivors at the facility. A safety distance of three kilometers for the human rescue team must be maintained. One of the remaining three units still operating at the reactor needs to be shutdown manually, as the units control system for automatically shutdown is not operating correctly due ti the explosion. The mission is to have an UAV find the building of the unit still operating and deploy a vehicle to enter the building. The UAV must carry visual sensors in order to find the right building, and obtain pictures of the panel gauges and switch positions in the control room, such that experts can assess the potential for a meltdown of the unit. The reconnaissance mission results in four levels.

In level 1 a flight over a distance of three kilometers with designated starting and final waypoint must be performed. During the flight the UAV must visit up to four waypoints. In level 2 the UAV must find a building entry indicated by a image consisting of a black circle with a white cross. In level 3 a vehicle must be deployed into the building, and image data, with sufficient quality for the judges to obtain the desired reconnaissance information, must be gathered. Level 4 performance is to complete level 1, level 2 and level 3 within 15 minutes. In each level the UAV must perform autonomous flight.

The work done by the aforementioned project groups and the Ph.d. student resulted i.a. in reconfiguring a Bergen Industrial Twin model helicopter (Helicopters [2000]) into an UAV platform by adding sensors, on-board computer, ground to helicopter communication and GPS equipment (see Figure 1.1). As much effort has been put into this rather comprehensive task, and the result is a fully functional and programmable UAV platform, this thesis takes its starting point here.

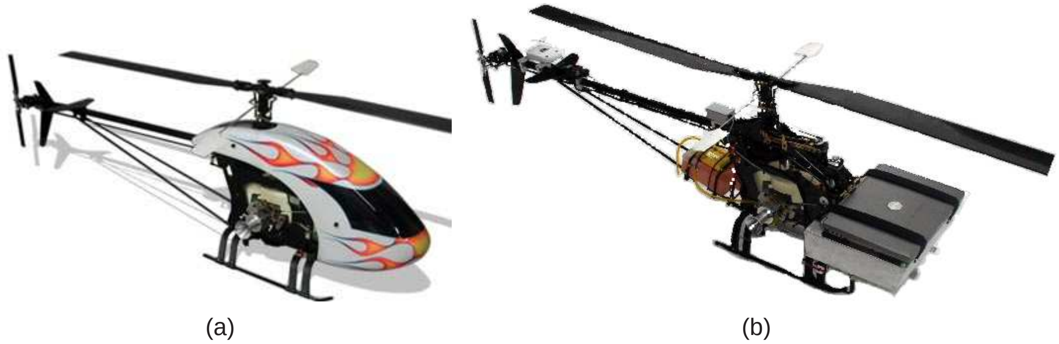


FIGURE 1.1: *The Bergen Industrial Twin model helicopter before (a) and after (b) being rebuild.*

The first of the above mentioned student projects (Hald et al. [2006]) concerned modelling of the Bergen helicopter using first principles. Some model parameter determination and Linear Quadratic Regulator (LQR) control were carried out as well. However, due to complications with the sensor equipment and much emphasis on the modelling task the groups never got to test the developed controllers in flight nor participate in the IARC 2006.

The complications with the sensor equipment has been taken care of by the supervisor of this master's thesis Anders la Cour-Harboe and the Ph.d. student Morten Bisgaard who is also working with the Bergen helicopter. In the Ph.d. project a nonlinear model of the Bergen helicopter and the necessary sensor fusion and estimation have been developed and implemented. Morten Bisgaard has made the sensor fusion and estimation and model implementation as well as his guidance available for the this thesis.

The second student project (Holmegaard et al. [2006]) concerned a navigation system enabling the UAV to complete level 1 and 2 in the IARC. The work done by the project group included development of a software platform and design of i.a. navigator, vision system, mission control, map generator and optimal path calculator. Tests of the navigation system showed the helicopter theoretically able to complete level 1 and 2. However, it is suggested to optimize the navigator by minimizing the flight time in level 1, by extending the autonomous flying features of the UAV to encompass fast forward flight also.

Inspired by the suggestion by Holmegaard et al. [2006] the overall objective of this master's thesis is stated as:

> *Design, implementation and test of a control strategy enabling the UAV to autonomously complete level 1 in the IARC by employing hover and fast forward flight.*

Here hover is defined as the UAV being airborne, and the translatory and rotational movement of the UAV are close to zero. And fast forward flight is defined as the UAV being airborne, and the rotational movement is close to zero, where a specified speed in the same direction as the heading of the UAV is maintained.

The fast forward feature is especially important, if a future project group at Aalborg University reaches the point where completion of the IARC, that is achieving level 4, becomes a goal. Because level 2 and 3 involves relaying reconnaissance data, and such tasks involves time-consuming image processing, the faster level 1 is completed the better. The overall objective is therefore rephrased as:

> *Design, implementation and test of different control strategies enabling the UAV to autonomously complete level 1 in the IARC by employing hover and fast forward flight, such that the most efficient control strategy regarding level 1 completion time can be identified.*

In order to accomplish the overall objective it has been chosen to look at the control system in Figure 1.2, from where the control strategies *linear control without gain scheduling* and *linear control with gain scheduling* can be seen. Based on the overall objective two linear controllers must be designed; one for hover and one for fast forward flight. The gain scheduling strategy is then implemented to use both of these linear controllers. The supervisory controller must handle the high level control of the system. The different blocks of the control system are elaborated on in the following.

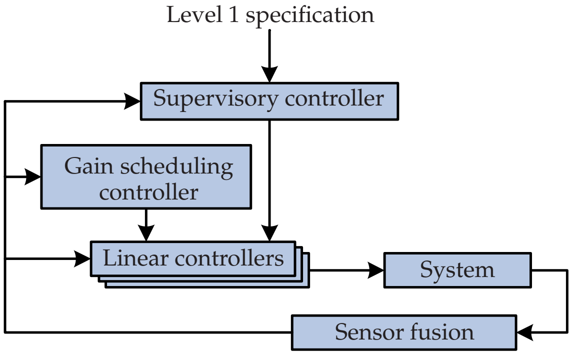


FIGURE 1.2: *Illustration of the elements constituting the control system. The arrows show the data flow direction.*

1.1 Prerequisites and Objectives

Using a bottom-up approach the blocks constituting the chosen control system are treated one by one throughout this thesis divided into a number of objectives. Prerequisites for achieving these objectives followed by the actual objectives are presented in the following. The order of these sections also serve as an outline for the chapters in part I and II of this thesis.

Prerequisite A: System and sensor fusion

As mentioned this thesis takes its starting point in the work done on modelling the reconfigured Bergen helicopter and development of sensor fusion and estimation. A nonlinear model (including sensor fusion and estimation) implemented in C++ for SIMULINK, able to simulate the motion of the Bergen helicopter, is available for this thesis. To narrow the focus of this thesis to the control field, this model is used for control design and validation tests of the developed strategies.

As model parameters, important for the nonlinear model to reflect the Bergen helicopter precisely, are being determined during the period of this thesis, it has been chosen not to perform test flights. However, as the complexity of the nonlinear model is assumed to be as high as for reflection of the Bergen helicopter, it is justified to use validations based on simulation for testing developed control strategies. Therefore, when referring to the UAV throughout the thesis it refers to the nonlinear model of the Bergen helicopter. An introduction to the nonlinear model and the sensor fusion and estimation can be found in Chapter 2.

Prerequisite B: Control system analysis

The control system will be analyzed with the purpose of identifying in- and outputs of the different blocks from Figure 1.2. This will lead to the identification of the needed references, feedback signals and controller outputs. The analysis can be found in Chapter 3

Prerequisite C: Control goals and requirements

It is necessary to determine control goal and requirements for the supervisory controller and linear controllers. These will be determined with respect to the overall objective stated earlier. These goals and requirements can be found in Chapter 4.

Objective A: Classic linear control

Based on the available feedback signal linear control, using classic Single Input Single Output (SISO) controllers, must be able to stabilize the UAV in hover. It provides a basis for comparison between SISO and Multiple Input Multiple Output (MIMO) control of the system. The design of classic linear control is found in Chapter 5.

Objective B: LQR hover control

An optimal LQR controller able to stabilize the UAV in hover must be developed. The performance of this controller is compared with the performance of the classic linear controller. Furthermore, it will be used as one of the two controllers in the gain scheduling strategy. Because the LQR controller is a MIMO controller, and the UAV is a MIMO system as well, it is expected to perform better.

Objective C: LQR forward flight control

In order to facilitate fast forward flight, a second optimal LQR controller must be developed for a forward flight operating point. The forward flight controller is expected to perform better than the LQR hover controller in the forward flight operating point. The design of the optimal LQR hover and forward flight controllers can be found in Chapter 6.

Objective D: Gain scheduling

A gain scheduling controller must be designed such it is possible to control the UAV utilizing the two optimal LQR controllers, such that an acceleration from hover to forward flight can be performed. The performance of this control method is evaluated by comparing it to the performance of the LQR hover and forward flight controllers. The design of gain scheduling control is found in Chapter 7.

Objective E: Supervisory controller

Based on a given level 1 specification defining the placement of waypoints, and position feedback from the UAV, a supervisory controller must calculate at which speed and heading the UAV must fly, in order to get through the given level 1 as fast as possible. The complexity of the supervisor (short for supervisory controller) is kept low, as the emphasis of this thesis is on linear control and gain scheduling. However, it is an important element in the control system in order to obtain a realistic estimate of the level 1 flight time for the Bergen helicopter platform. Design of the supervisor can be found in Chapter 8.

1.2 Control system testing and Conclusions

When the above listed objectives have been reached, the designed control strategies will be tested with respect to the overall objective and the IARC. Because the range of this thesis only span as far as level 1 of the IARC the designed control strategies will only be tested with respect to the criteria listed for this level. The control system test can be found in Chapter 9.

Finally the thesis will sum up the concluded results obtained from the different objectives . The conclusions ultimately leads to suggestions for future work to be done regarding the Bergen helicopter. The conclusion and future work is found in Chapter 10.

Part I

Helicopter model introduction and control system analysis

2
Helicopter model introduction

To be able to perform satisfactory control of the helicopter it is necessary to obtain knowledge about the model. This includes knowledge about the different reference frames, the general model structure, and the inputs and outputs of the model.

2.1 Reference frames

For the purpose of performing control of the helicopter two reference frames are used; an Earth-fixed reference Frame (EF) and a Body-fixed reference Frame (BF). The xy-plane of the EF is parallel to the surface of the Earth, where the x-axis is pointing north, the y-axis is pointing east and the z-axis perpendicular to both pointing vertically downwards. The origin of the EF is chosen arbitrarily but always with the aforementioned orientation. The EF is used to describe the position and attitude of the helicopter.

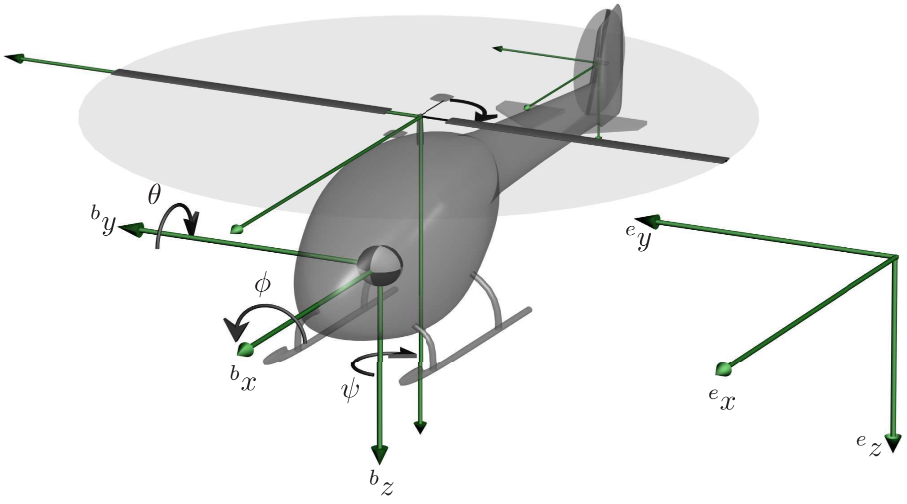


Figure 2.1: *Illustration of the two coordinate systems used for control purposes.*

The second reference frame necessary for control purposes is the BF, which has its origin in the center of mass of the helicopter, and follows the position and attitude of the helicopter. The x-axis of the BF is pointing through the nose of the helicopter, the y-axis point through the right side of the helicopter and the z-axis perpendicular to both and pointing downwards through the bottom of the helicopter. It is in the BF that translateral and rotational movement of the helicopter are defined. The reference frames are illustrated in Figure 2.1. The illustration was made by

Hald et al. [2006], whom used the two additional frames attached to the main and tail rotor for modeling purposes. These additional frames will not considered in this thesis.

2.2 Model overview

To perform control of the nonlinear model only the input/output relations are taken into consideration. Therefore, the actual model components are looked upon as black boxes, which receive some actuator inputs and yield some outputs regarding the position, attitude and movement. The description of the black boxes will be strictly superficial, where only the build-up and functionality will be elaborated on.

The model used is divided into three black boxes; the nonlinear helicopter model, the sensor emulation, and the sensor fusion and estimation. The three blocks can be seen in Figure 2.2.

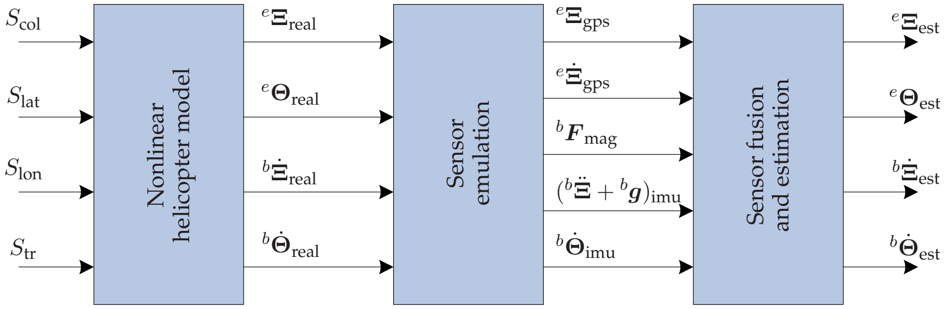

S_{col}
S_{lat}
S_{lon}
S_{tr}
Nonlinear helicopter model
${}^e\boldsymbol{\Xi}_{\text{real}}$
${}^e\boldsymbol{\Theta}_{\text{real}}$
${}^b\dot{\boldsymbol{\Xi}}_{\text{real}}$
${}^b\dot{\boldsymbol{\Theta}}_{\text{real}}$
Sensor emulation
${}^e\boldsymbol{\Xi}_{\text{gps}}$
${}^e\dot{\boldsymbol{\Xi}}_{\text{gps}}$
${}^b\boldsymbol{F}_{\text{mag}}$
$({}^b\ddot{\boldsymbol{\Xi}} + {}^b\boldsymbol{g})_{\text{imu}}$
${}^b\dot{\boldsymbol{\Theta}}_{\text{imu}}$
Sensor fusion and estimation
${}^e\boldsymbol{\Xi}_{\text{est}}$
${}^e\boldsymbol{\Theta}_{\text{est}}$
${}^b\dot{\boldsymbol{\Xi}}_{\text{est}}$
${}^b\dot{\boldsymbol{\Theta}}_{\text{est}}$

FIGURE 2.2: *Block diagram of the model used in this thesis illustrating the nonlinear helicopter model, the sensor emulation, and the sensor fusion and estimation. All with related inputs and outputs.*

Nonlinear helicopter model: The nonlinear model of the helicopter is constructed by several mathematical equations describing how the helicopter acts when given any arbitrary input. These mathematical equations include a dynamic model of the actuators mounted on the helicopter, a dynamic model of the flapping motion of the main rotor blades and the stabilizer bar, a calculation of the forces and torques generated by and affecting the helicopter and a model of the rigid body dynamics and kinematics, which the forces and torques are acting upon. As it can be seen from Figure 2.2 the nonlinear model takes four inputs; one for the collective pitch of the main rotor blades S_{col}, one for the lateral cyclic pitch angle of the main rotor S_{lat}, one for the longitudinal cyclic pitch of the main rotor S_{lon} and one defining the reference for the build-in yaw rate control of the tail rotor S_{tr}. Based on the value of the inputs the nonlinear model yields four outputs for control purposes; the position given in the EF ${}^e\boldsymbol{\Xi}_{\text{real}}$, the attitude given in the EF ${}^e\boldsymbol{\Theta}_{\text{real}}$, the translatory velocity given in the BF ${}^b\dot{\boldsymbol{\Xi}}_{\text{real}}$ and the angular velocity given in the BF ${}^b\dot{\boldsymbol{\Theta}}_{\text{real}}$.

Sensor emulation: The sensor emulation block contains models of the different sensors mounted on the helicopter. These sensor models have had measurement noise added to them, where the noise has been determined based on tests performed on the actual helicopter. Note that the information provided by the sensors (both actual and emulation) are not sufficient for control purposes, hence sensor fusion and estimation is necessary. The sensors used for measurements on the helicopter are a GPS, which measures the position ${}^{e}\boldsymbol{\Xi}_{\text{gps}}$ and translatory velocity ${}^{e}\dot{\boldsymbol{\Xi}}_{\text{gps}}$ in the EF, a compass that measures the attitude of the helicopter with respect to the Earths magnetic field ${}^{b}\boldsymbol{F}_{\text{mag}}$ and an IMU, which measures the the sum of the gravitational and helicopter accelerations $({}^{b}\ddot{\boldsymbol{\Xi}} + {}^{b}\boldsymbol{g})_{\text{imu}}$ and the angular velocities ${}^{b}\dot{\boldsymbol{\Theta}}_{\text{imu}}$.

Sensor fusion and estimation: The sensor fusion and estimation blocks take the information from the sensor emulations and utilizes them to estimate the position, attitude and velocities of the nonlinear model. The estimator is designed as an *Unscented Kalman Filter*, which is statistical calculation method of a random variable undergoing a nonlinear transformation. The outputs of the estimator are the estimated states of the nonlinear model; the position ${}^{e}\boldsymbol{\Xi}_{\text{est}}$, the attitude ${}^{e}\boldsymbol{\Theta}_{\text{est}}$, the translatory velocity ${}^{b}\dot{\boldsymbol{\Xi}}_{\text{est}}$ and the angular velocity ${}^{b}\dot{\boldsymbol{\Theta}}_{\text{est}}$.

From the above description of the model (used in this thesis) the input and output vectors can be determined as

$$\boldsymbol{u} = \begin{bmatrix} S_{\text{col}} & S_{\text{lat}} & S_{\text{lon}} & S_{\text{tr}} \end{bmatrix}^{T} \tag{2.1}$$

$$\begin{aligned} \boldsymbol{x} &= \begin{bmatrix} {}^{e}\boldsymbol{\Xi}^{T} & {}^{e}\boldsymbol{\Theta}^{T} & {}^{b}\dot{\boldsymbol{\Xi}}^{T} & {}^{b}\dot{\boldsymbol{\Theta}}^{T} \end{bmatrix}^{T} \\ &= \begin{bmatrix} {}^{e}x & {}^{e}y & {}^{e}z & {}^{e}\phi & {}^{e}\theta & {}^{e}\psi & {}^{b}\dot{x} & {}^{b}\dot{y} & {}^{b}\dot{z} & {}^{b}\dot{\phi} & {}^{b}\dot{\theta} & {}^{b}\dot{\psi} \end{bmatrix}^{T}, \end{aligned} \tag{2.2}$$

where $\boldsymbol{S}$ is the inputs to the system and $\boldsymbol{x}$ are the outputs of the system to be controlled. Note that the output vector $\boldsymbol{x}$ is the output of the estimator, and that the subscript est is removed to ease further use of the output signals.

2.3 Input/output relations

To ease the design of different controllers for the model described above the input/output relations are analyzed further. In addition, to ease the analysis of the relations the effect of cross couplings between model states are not taken into consideration. Note that the descriptions below are based on the BF having the exact same orientation as the EF.

Collective input S_{col}: This input controls the collective pitch of the main rotor blades, which makes the helicopter move vertically. Therefore, this input affects the vertical position ${}^{e}z$ and the vertical velocity ${}^{b}\dot{z}$.

Cyclic input S_{lat}: This input controls the lateral pitch angle of the main rotor blades making the helicopter move sideways. This means that this input affect the lateral position ${}^{e}y$, the roll angle of the helicopter ${}^{e}\phi$, the lateral velocity ${}^{b}\dot{y}$ and the roll angular velocity ${}^{b}\dot{\phi}$.

Cyclic input S_{lon}: This input controls the longitudinal pitch of the main rotor blades, which makes the helicopter move for- or backwards. As a result it is determined that this input affect the longitudinal position ${}^{e}x$, the pitch angle of the helicopter ${}^{e}\theta$, the longitudinal translatory velocity ${}^{b}\dot{x}$ and the pitch angular velocity ${}^{b}\dot{\theta}$.

Reference input S_{tr}: This input sets the reference for the build-in yaw rate controller. This means that this input ultimately affects the yaw angle of the helicopter ${}^{e}\psi$ and the yaw angular velocity ${}^{b}\dot{\psi}$.

The descriptions of the four inputs can in association with the knowledge gained from the reference frames be summarized as listed in Table 2.1. Note that the positions and angles given in the EF are left out because the BF does not have the same orientation as the EF at all times.

Input	Translatory movement	Rotational movement
$S_{\text{col}} > 0$	${}^{b}\dot{z} > 0$	–
$S_{\text{col}} < 0$	${}^{b}\dot{z} < 0$	–
$S_{\text{lat}} > 0$	${}^{b}\dot{y} < 0$	${}^{b}\dot{\phi} < 0$
$S_{\text{lat}} < 0$	${}^{b}\dot{y} > 0$	${}^{b}\dot{\phi} > 0$
$S_{\text{lon}} > 0$	${}^{b}\dot{x} > 0$	${}^{b}\dot{\theta} < 0$
$S_{\text{lon}} < 0$	${}^{b}\dot{x} < 0$	${}^{b}\dot{\theta} > 0$
$S_{\text{tr}} > 0$	–	${}^{b}\dot{\psi} > 0$
$S_{\text{tr}} < 0$	–	${}^{b}\dot{\psi} < 0$

TABLE 2.1: *Overview of the collective, cyclic and reference inputs to the helicopter describing the effects of a given input with respect to its sign. This effect is seen for the system initialized in hover and only one input applied at a time.*

2.4 The feedback signal

It is important to keep in mind that the feedback signals from the estimator are not perfectly good signals, hence it may be necessary to design controllers a bit slow to be able to handle any form of sudden alteration in the signal. In addition, some estimated states are more affected by the measurement noise from the sensors. An example of this can be seen in Figure 2.3, which shows the roll, pitch and yaw angles (${}^{e}\phi$, ${}^{e}\theta$ and ${}^{e}\psi$ respectively) as model states (to the left) and as estimated states (to the right) for the nonlinear model trimmed in hover. It is observed, that the yaw angle (red line) is the least correct estimate of the three. In addition, the roll angle (blue line) is somewhat affected by the estimation of the yaw angle, which is indicated by the way the roll angle seems to follow changes in the yaw angle estimate. In Appendix B graphs of all 12 states are showed; both real and estimated states for comparison.

Having determined the reference frames for control purposes, and the inputs and outputs of

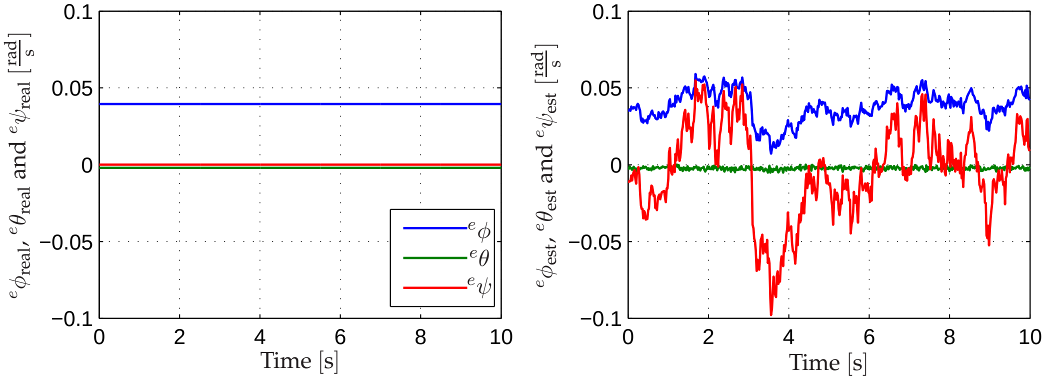


FIGURE 2.3: *Comparison of the real roll, pitch and yaw angles (left) and the estimated roll, pitch and yaw angles (right). The data is obtained by initializing the system in hover. Note that the legend on the left graph is also applicable on the right graph.*

the system model it is now possible to analyze the control system further, which will be done in the following chapter.

3

Control system analysis

This chapter describes an analysis of the physical environment in level 1 of the IARC with regard to the overall goal stated in the introduction. Then an overview of the different control units are presented, and finally the control system elaborated on in this thesis is established.

3.1 The physical environment in level 1

As the level 1 specification in the IARC is unknown, the UAV must be able to handle an infinitely number of different shapes specified by GPS coordinates. Examples on flight paths are illustrated in Figure 3.1.

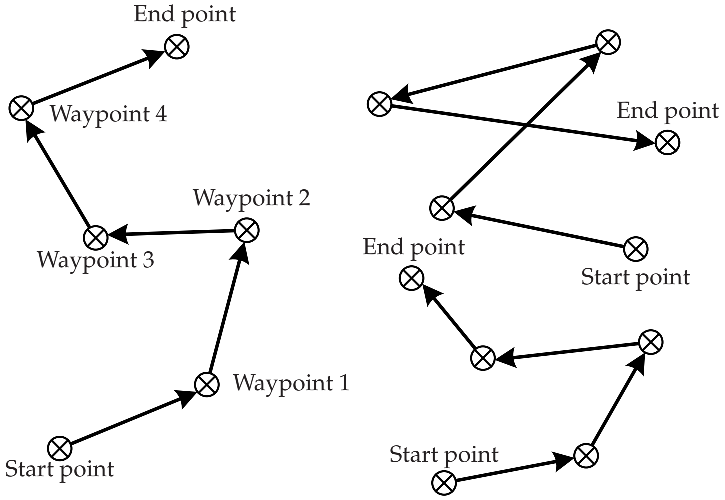


FIGURE 3.1: *Three examples on flight paths the UAV might be subject to in level 1 of the IARC. The scaling of the figure should be ignored, as the total length of each flight path is known to be three kilometers.*

From the figure it can be seen, that both lengths on the path segments and the turn angles between the path segments might vary significantly from waypoint to waypoint as well as for different level 1 specifications. As described in the introduction it is desired to travel the total flight path of level 1 as fast as possible, hence the UAV must be able to calculate a time-optimizing way of dealing with a given set of path segments and waypoint turns.

When considering optimization regarding the flight path of the UAV, this can be done by optimizing the flight through the different waypoints defined for a given level 1 specification with respect to speed and turning method. A likely approach to this problem is to implement an optimization algorithm in a supervisory controller in this thesis. Note that there are several different ways of dealing with different waypoints with respect to turning, speed and heading.

However, since the focus of this thesis is on designing linear controllers and gain scheduling it is chosen only to look at two different types of dealing with waypoints; decelerating to hover and turn, or fly over a waypoint with a given speed and then turn maintaining that same speed.

With the above analysis of the physical environment of level 1 in mind it is possible to elaborate further on the defined control system from the introduction (see Figure 1.2 on page 3).

3.2 Control system overview

The general control system to be developed in this thesis consists of two main control blocks; a supervisor and a linear controller block (the general control system in open loop is illustrated in Figure 3.2). Because gain scheduling per definition consists of two or more controllers it can not be seen as a direct block in the control system, but rather a frame work around several controllers, which will be elaborated on later.

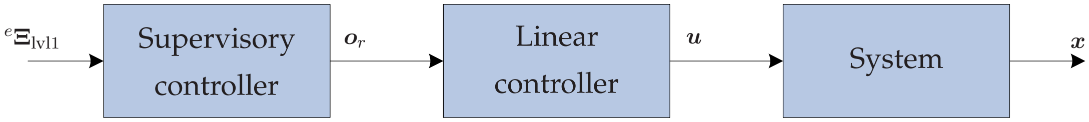


FIGURE 3.2: *Block diagram of the general control system illustrated in open loop with in- and outputs of each block.*

In the following the two controller blocks from Figure 3.2 will be elaborated on.

3.2.1 Supervisory controller

As it can be seen from Figure 3.2 the high level control of the control system is handled by the supervisor. As described earlier the supervisor must be able to define the optimal flight path for the UAV based on the information received about the waypoint positions of level 1 ${}^{e}\mathbf{\Xi}_{\mathrm{lvl1}}$. Furthermore, the supervisor must be able to calculate references $\boldsymbol{o}_r$ for the linear controller, which handles the low level control of the control system. Based on this information about the reference input and the output of the supervisor a more clear definition of the supervisor regarding its actual functionality can be determined.

Since the supervisor handles the planning of the actual flight path for the UAV it is chosen to have the supervisor handle the entire position control of the control system. In addition, since the supervisor determines the method as to how the UAV must fly through the different waypoints it is also necessary for the supervisor to know the heading of the UAV. This result in the feedback signal vector for the supervisor to be

$$\begin{bmatrix} {}^{e}\mathbf{\Xi}^{T} & {}^{e}\psi \end{bmatrix}^{T} , \tag{3.1}$$

where ${}^{e}\mathbf{\Xi}$ is the position vector given as $\begin{bmatrix} {}^{e}x & {}^{e}y & {}^{e}z \end{bmatrix}$. This leads to the in illustration of the supervisor block (see Figure 3.3).

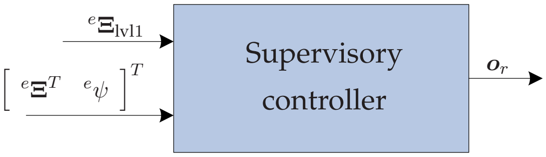

${}^{e}\Xi_{\mathrm{lvl1}}$
$\begin{bmatrix} {}^{e}\Xi^T & {}^{e}\psi \end{bmatrix}^T$
Supervisory controller
$\boldsymbol{o}_r$

FIGURE 3.3: *Illustration of the in- and outputs for the supervisor.*

Based on the inputs the supervisor must be able to calculate references $\boldsymbol{o}_r$ for the linear controller. Since the supervisor itself handles the position control of the UAV it is not necessary to calculate references for these states. As mentioned in Chapter 2 the total state vector is

$$\boldsymbol{x} = \begin{bmatrix} {}^{e}x & {}^{e}y & {}^{e}z & {}^{e}\phi & {}^{e}\theta & {}^{e}\psi & {}^{b}\dot{x} & {}^{b}\dot{y} & {}^{b}\dot{z} & {}^{b}\dot{\phi} & {}^{b}\dot{\theta} & {}^{b}\dot{\psi} \end{bmatrix}^T .$$

By removing the position states from the state vector there are a total of nine states left, which are all desired to be able to control. Therefore, the output reference vector from the supervisor must calculate references for these nine states, which yields the output reference vector

$$\boldsymbol{o}_r = \begin{bmatrix} r_{{}^{e}\phi} & r_{{}^{e}\theta} & r_{{}^{e}\psi} & r_{{}^{b}\dot{x}} & r_{{}^{b}\dot{y}} & r_{{}^{b}\dot{z}} & r_{{}^{b}\dot{\phi}} & r_{{}^{b}\dot{\theta}} & r_{{}^{b}\dot{\psi}} \end{bmatrix}^T . \tag{3.2}$$

3.2.2 Linear controller

Having determined the reference input to the linear controller the control states becomes

$$\begin{bmatrix} {}^{e}\boldsymbol{\Theta}^T & {}^{b}\dot{\boldsymbol{\Xi}}^T & {}^{b}\dot{\boldsymbol{\Theta}}^T \end{bmatrix}^T = \begin{bmatrix} {}^{e}\phi & {}^{e}\theta & {}^{e}\psi & {}^{b}\dot{x} & {}^{b}\dot{y} & {}^{b}\dot{z} & {}^{b}\dot{\phi} & {}^{b}\dot{\theta} & {}^{b}\dot{\psi} \end{bmatrix}^T . \tag{3.3}$$

It is not necessary to use all of the feedback states, which means that the linear controller may designed to only utilize some of the feedback states and references. However, a number of four states are the absolute minimum needed to perform control of the UAV, where these states are the translateral velocities and the yaw angle:

$$\begin{bmatrix} {}^{e}\psi & {}^{b}\dot{x} & {}^{b}\dot{y} & {}^{b}\dot{z} \end{bmatrix}^T . \tag{3.4}$$

As described in the introduction to this thesis the linear controller either consist of classic SISO control or MIMO optimal LQR control, where it is necessary to design one controller for each of the feedback signals when using SISO control, and only one controller for all feedback states when using LQR control.

Based on the above the actual linear controller block can be defined with respect to inputs and output (see Figure 3.4). The linear controller receives the reference signal $\boldsymbol{o}_r$ from the supervisor along with the above described feedback signal $[\ {}^{e}\boldsymbol{\Theta}^T \ \ {}^{b}\dot{\boldsymbol{\Xi}}^T \ \ {}^{b}\dot{\boldsymbol{\Theta}}^T\ ]^T$ as inputs. The output of the linear controller is then the input vector $\boldsymbol{u}$ to the system.

3.2.3 Closed loop control system

Having described in- and outputs of both the supervisor and the linear controller the control system layout shown in Figure 1.2 on page 3 can be transformed into the closed loop control

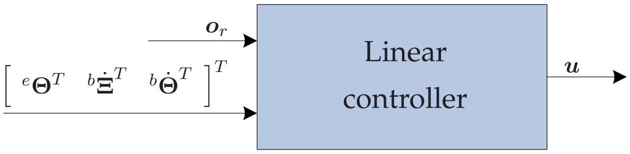

$\boldsymbol{o}_r$
$\left[\begin{array}{ccc} {}^{e}\boldsymbol{\Theta}^T & {}^{b}\dot{\boldsymbol{\Xi}}^T & {}^{b}\dot{\boldsymbol{\Theta}}^T \end{array}\right]^T$
Linear controller
$\boldsymbol{u}$

FIGURE 3.4: *Illustration of the in- and outputs for a linear controller, which is either classic hover, LQR hover or LQR forward flight controller.*

system in Figure 3.5.

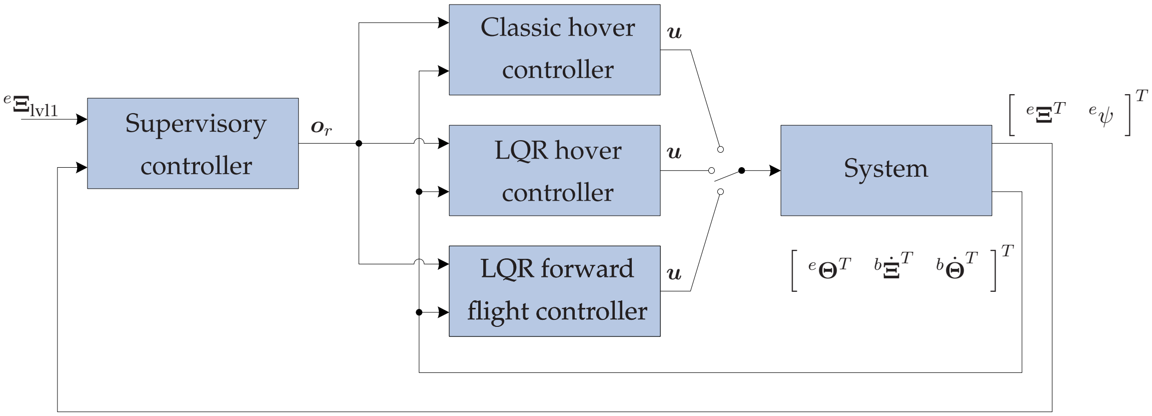

${}^{e}\boldsymbol{\Xi}_{\mathrm{lvl1}}$
Supervisory controller
$\boldsymbol{o}_r$
Classic hover controller
$\boldsymbol{u}$
LQR hover controller
$\boldsymbol{u}$
LQR forward flight controller
$\boldsymbol{u}$
System
$\left[\begin{array}{cc} {}^{e}\boldsymbol{\Xi}^T & {}^{e}\psi \end{array}\right]^T$
$\left[\begin{array}{ccc} {}^{e}\boldsymbol{\Theta}^T & {}^{b}\dot{\boldsymbol{\Xi}}^T & {}^{b}\dot{\boldsymbol{\Theta}}^T \end{array}\right]^T$

FIGURE 3.5: *Illustration of the closed loop control system using only linear control. The linear controllers are designed separately and can be utilized one at the time, as indicated by the switch at the system input.*

As it can be seen from Figure 3.5 each of the linear controllers can be used one at the time. As for the general illustration of the control system, gain scheduling is left out but will be elaborated on in the following section.

3.2.4 Closed loop control system using gain scheduling

The gain scheduling approach used in this project is based on Bendtsen et al. [2005]. The two LQR controllers in Figure 3.5 are inserted in an observer based control structure, and using a variable $\alpha \in [0; 1]$, the influence of the control signals from the LQR controllers are each weighted as illustrated in Figure 3.6.

Having determined the structure of the control system along with the inputs and outputs of the different controller blocks, the requirements and test specification for the control system can be identified.

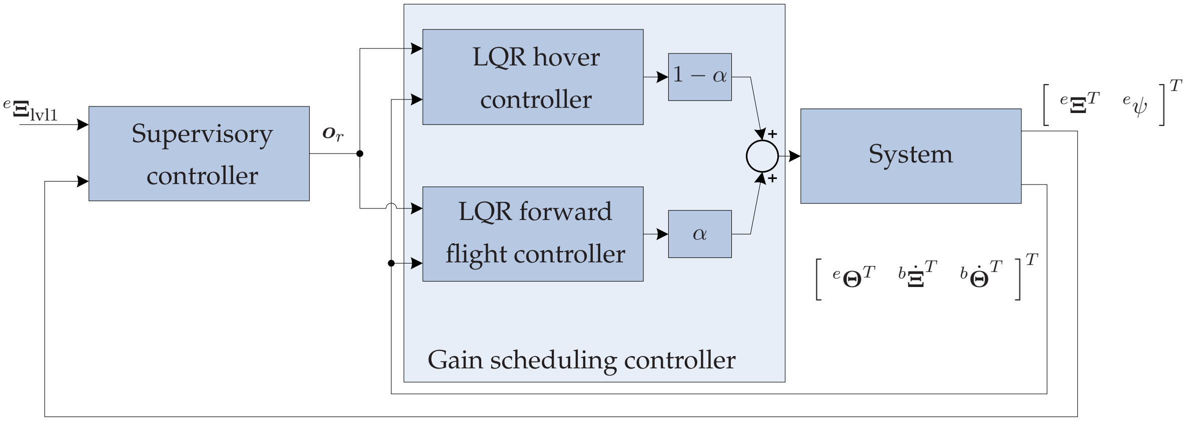

${}^{e}\boldsymbol{\Xi}_{\text{lvl1}}$
Supervisory controller
$\boldsymbol{o}_r$
LQR hover controller
$1-\alpha$
LQR forward flight controller
α
+
+
System
$\begin{bmatrix} {}^{e}\boldsymbol{\Xi}^T & {}^{e}\psi \end{bmatrix}^T$
$\begin{bmatrix} {}^{e}\boldsymbol{\Theta}^T & {}^{b}\dot{\boldsymbol{\Xi}}^T & {}^{b}\dot{\boldsymbol{\Theta}}^T \end{bmatrix}^T$
Gain scheduling controller

FIGURE 3.6: *Illustration of the gain scheduling approach used in this thesis. The classic controller block is left out, as it is not used for the gain scheduling control.*

4
Control goals and requirements

This chapter describes how the requirements to the elements constituting the control system have been established. Hence the overall requirements and settings for the supervisory controller are explained, together with the requirements for the linear controllers, which primary concern the validation of the controllers.

4.1 Supervisory controller

To setup requirements to the control system elements starting point is taken in the overall objective described in the introduction *to complete the level 1 of the IARC*. A review of the rules associated with the competition specifies that an autonomous UAV must be able to perform a flight of three kilometers visiting up to four waypoints. Whether the waypoints have been visited is determined by judges placed on the ground at each waypoint. If the judge observes the UAV when looking up the waypoint has been passed. It is assumed that passing the waypoints within a circle with a radius of two meters, is sufficient for the judges to observe the UAV. As the position control is performed by the supervisor, this is a requirement for this controller. In addition, the UAV must initiate and terminate the level 1 flight in hover. Also a requirement for the supervisor. These are the only control requirements given in advance and determined by AUVSI [1972]. However, to obtain steady hover and forward flight goals and requirements to the linear controllers must be established as well.

4.2 Linear controllers

The requirements described in this section concern the control states in (3.3) and some overall properties necessary for the control system.

The overall objective repeated above, obviously, results in the need for a control system able to bring the UAV from hover to forward flight and back to hover again. Therefore, the linear controllers must be able to follow the references calculated by the supervisor. In addition, the controllers must feature integral states to avoid possible steady state errors.

The controllers must stabilize the system using the rather noisy state estimates discussed in Section 2.4 as feedback. The controlled states can therefore be expected to fluctuate from their respective references. As a control goal these fluctuations must be minimized in order to obtain steady control.

As a requirement to the dynamical properties the controllers must be able to stabilize the UAV, when the UAV is initialized in a condition not equal to the operating point for the given controller. This condition is therefore described by offsets, which are added to the initial values, in translateral velocities and angles with respect to an operating point. The offsets values listed in Table 4.1 have been chosen.

State:	${}^{e}\phi$	${}^{e}\theta$	${}^{e}\psi$	${}^{b}\dot{x}$	${}^{b}\dot{y}$	${}^{b}\dot{z}$
Offset value:	$0.5\,\mathrm{rad}$	$0.5\,\mathrm{rad}$	$0.5\,\mathrm{rad}$	$2\,\frac{\mathrm{m}}{\mathrm{s}}$	$2\,\frac{\mathrm{m}}{\mathrm{s}}$	$2\,\frac{\mathrm{m}}{\mathrm{s}}$

TABLE 4.1: *Offset values specifying the rquirements to the dynammical properties of the linear controllers.*

The following sums up the requirements established in this section.

Supervisory controller

- The supervisor must be able convert a given level 1 specification to references for use by the linear controllers, such that the UAV can complete the level 1 path autonomously.
- Waypoints must be passed within a circle with a radius of $2\,\mathrm{m}$ in the xy-plane.
- The level 1 flight must be initiated and terminated in hover.

Linear controller

- The linear controllers must feature integral states to avoid steady state errors.
- Fluctuation on the controlled states must be minimized, in order to obtain steady control.
- The controllers must be able to stabilize the UAV, when the UAV initialized with the offset values in Table 4.1.

Having established the desired control system structure in the previous chapter, and the control goals and requirements above, the next chapter describes the design the classic linear controller.

Part II

Control system design

5 Classic linear control

In this chapter the design and implementation of a classic linear SISO controller structure will be described. Since there are several different methods of designing SISO controllers the method used in this thesis will be described initially. Note that there will only be designed a SISO control system capable of keeping the UAV in hover.

5.1 Method overview

The actual model of the Bergen helicopter is, as mentioned in Chapter 2, nonlinear, why linear control using SISO controllers is rather difficult. Therefore, the nonlinear model is linearized in the hover operating point (no movement), where there will be designed linear controllers for this specific flight condition. The performance of the linear SISO controllers will ultimately be used for comparison with the more advanced control method LQR, which will be elaborated on later in this thesis.

Since a linearized model of the nonlinear model is available it is possible to design the controllers based on analysis of the transfer functions of the linearized models obtained from the system matrices (see Appendix C). Based on the transfer functions, it is possible to identify poles and zeros and to analyze the stability of the system with respect to the different outputs, which is done using root locus plots. The use of root locus plots to design the SISO controllers leads to the determination of lead or lag compensators, which are approximations of PI and PD controllers, but can be just as effective.

The initial design of the controllers will be done based on the actual model feedback to ease the identification of stability using the designed controllers. If stability is obtained using model feedback, then the controllers are tested using the estimator states as feedback, which helps identify the robustness of the designed controllers.

5.2 Controller structure

Because the classic control strategy is based on SISO controllers it is necessary to design several controllers; one for each of the desired outputs to be controlled. In Chapter 2 it was determined that the total state vector available for control purposes has a total of 12 states. Since the overall control strategy described in Chapter 3 states that the position control of the total control system is to be handled by a supervisor, only nine states are left to use as feedback. In addition, it is not deemed necessary to design SISO controllers for the angular velocities of the helicopter, which leads to the six remaining states used for feedback:

$$\boldsymbol{x}_{\text{fb}} = \begin{bmatrix} {}^e\boldsymbol{\Theta}^T & {}^b\dot{\boldsymbol{\Xi}}^T \end{bmatrix}^T = \begin{bmatrix} {}^e\phi & {}^e\theta & {}^e\psi & {}^b\dot{x} & {}^b\dot{y} & {}^b\dot{z} \end{bmatrix}^T \quad . \tag{5.1}$$

This result in a total of six controllers to be designed in the classic linear control system, where the controller structure is illustrated in Figure 5.1. The controllers are denoted $\mathcal{C}_i$ with i being the indicator for which state is being controlled.

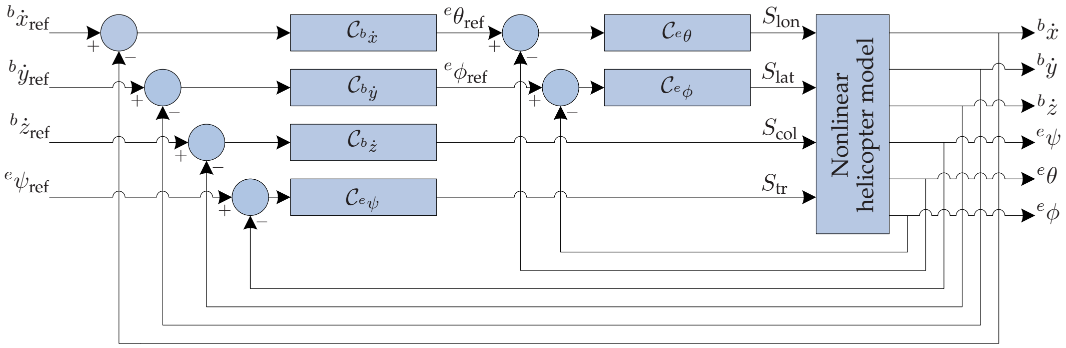


FIGURE 5.1: *Illustration of the controller structure used for classic linear control of the system. As can be seen there are implemented cascade couplings for longitudinal and lateral movement.*

With the structure of the total classic linear control system defined it is possible to design the different controllers. From Figure 5.1 it can be seen that $\mathcal{C}_{{}^b\dot{x}}$ and $\mathcal{C}_{{}^b\dot{y}}$ must be designed such that their outputs are references to $\mathcal{C}_{{}^e\theta}$ and $\mathcal{C}_{{}^e\phi}$ respectively. This means that for the purpose of correct controller design $\mathcal{C}_{{}^e\theta}$ and $\mathcal{C}_{{}^e\phi}$ must be designed before $\mathcal{C}_{{}^b\dot{x}}$ and $\mathcal{C}_{{}^b\dot{y}}$. The two remaining controllers $\mathcal{C}_{{}^b\dot{z}}$ and $\mathcal{C}_{{}^e\psi}$ can be designed without any prerequisite.

5.3 Controller design

The design of the classic linear controllers is divided into four overall design parts; longitudinal control, lateral control, vertical control and yaw control. The controllers will be validated as they have been designed. Furthermore, a complete test of the total classic linear control system will be performed when all the controllers have been designed.

5.3.1 Design procedure

Throughout this section the controllers will be designed based on reduced versions of the actual linearized state-space model of the nonlinear model by common procedure. The reduced state-space models will be shown as standard state-space models:

$$\begin{aligned} \dot{\boldsymbol{x}}_i &= \boldsymbol{A}_i\boldsymbol{x}_i + \boldsymbol{B}_i\boldsymbol{u} \\ \boldsymbol{y}_i &= \boldsymbol{C}_i\boldsymbol{x}_i + \boldsymbol{D}_i\boldsymbol{u} \quad , \end{aligned}$$

where i indicating the controlled state. Furthermore, the transfer functions for the different controlled states are calculated based on their state-space models using the following operation:

$$G_i = \boldsymbol{C}_i \left(\mathrm{s}\boldsymbol{I} - \boldsymbol{A}_i \right)^{-1} \boldsymbol{B}_i + \boldsymbol{D}_i \ \ . \tag{5.2}$$

With the determination of the transfer function for the control states the different controllers can be designed using root locus plots.

5.3.2 Longitudinal control design

The first step in designing the longitudinal control part is to design the ${}^{e}\theta$ controller, where the first step of the design procedure is to determine the input/output transfer function $G_{{}^{e}\theta} = \frac{{}^{e}\theta}{S_{\text{lon}}}$. Based on the system matrices of the linearized model, the state-space system specific for ${}^{e}\theta$ is determined as

$$\begin{bmatrix} {}^{e}\dot{\theta} \\ {}^{b}\ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0.9992 \\ 0 & -7.0002 \end{bmatrix} \begin{bmatrix} {}^{e}\theta \\ {}^{b}\dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ -42.5413 \end{bmatrix} S_{\text{lon}}$$

$$\boldsymbol{y}_{{}^{e}\theta} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} {}^{e}\theta \\ {}^{b}\dot{\theta} \end{bmatrix} + 0\, S_{\text{lon}} \ \ .$$

From the reduced state-space model it can be seen that all other states than ${}^{e}\theta$ and ${}^{b}\dot{\theta}$ have been decoupled to remove cross couplings, which helps ease the design of the controller.

From the state-space model and (5.2) it is possible to determine the transfer function for the system $G_{{}^{e}\theta}$ as

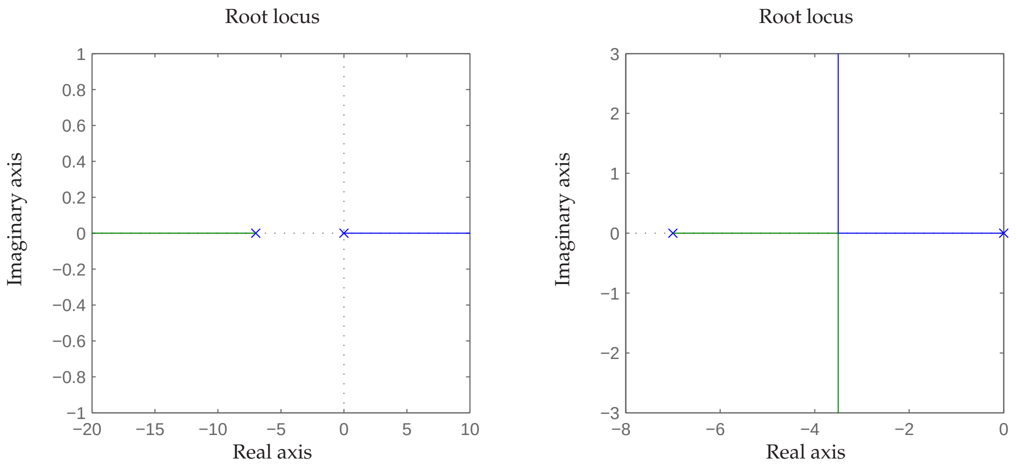
$$G_{{}^{e}\theta} = \frac{-42.51}{\mathrm{s}(\mathrm{s}+7)} \ \ . \tag{5.3}$$

When analyzing the root locus plot for $G_{{}^{e}\theta}$ it is seen that the root locus of the pole $\mathrm{s} = 0$ lies in the right half plane, which indicates that the system is unstable (see Figure 5.2(a)). Based on this observation it can be determined that the gain of the ${}^{e}\theta$ controller must be negative, which will causethe root locus of the aforementioned pole to be in the left half plane of the s-domain (see Figure 5.2(b)). Ultimately, it is possible to control ${}^{e}\theta$ by the use of a P-controller. However, the utilization of this type of controller may cause the system to have a steady state error. The pole and zero of the compensator is determined by using the MATLAB© toolbox `sisotool`, which immediately shows the effect of placing poles and zeros in the open loop system $\mathcal{C}_{{}^{e}\theta} G_{{}^{e}\theta}$. This result in a pole-zero placement for the compensator as

$$\mathcal{C}_{{}^{e}\theta} = -K_{p,{}^{e}\theta} \frac{\mathrm{s}+4}{\mathrm{s}+10} \ \ , \tag{5.4}$$

which yields the root locus plot of the open loop system for ${}^{e}\theta$ as shown in Figure 5.3.

From the root locus plot of the open loop system, it is possible to determine the actual control gain needed to perform stable control. The gain is chosen with the intend of having a closed loop



(a) (b)

FIGURE 5.2: *(a) shows the root locus plot of $G_{{}^e\theta}$. (b) shows the root locus plot of $G_{{}^e\theta}$ with negative control gain.*

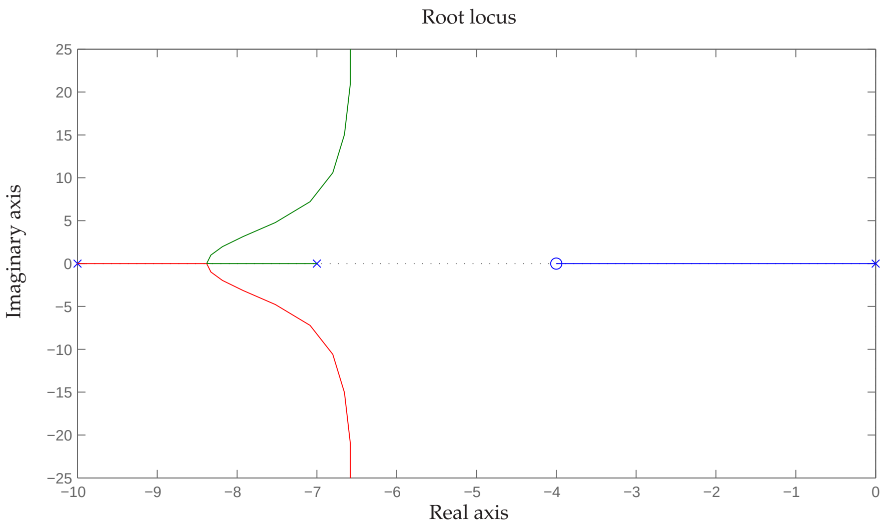


FIGURE 5.3: *Root locus plot of the open loop system $\mathcal{C}_{{}^e\theta}G_{{}^e\theta}$. From this plot the controller gain can be determined.*

steady state gain (DC gain) of 1 (0 dB), which result in a control gain of

$$K_{p,{}^e\theta} = 2.54 \tag{5.5}$$

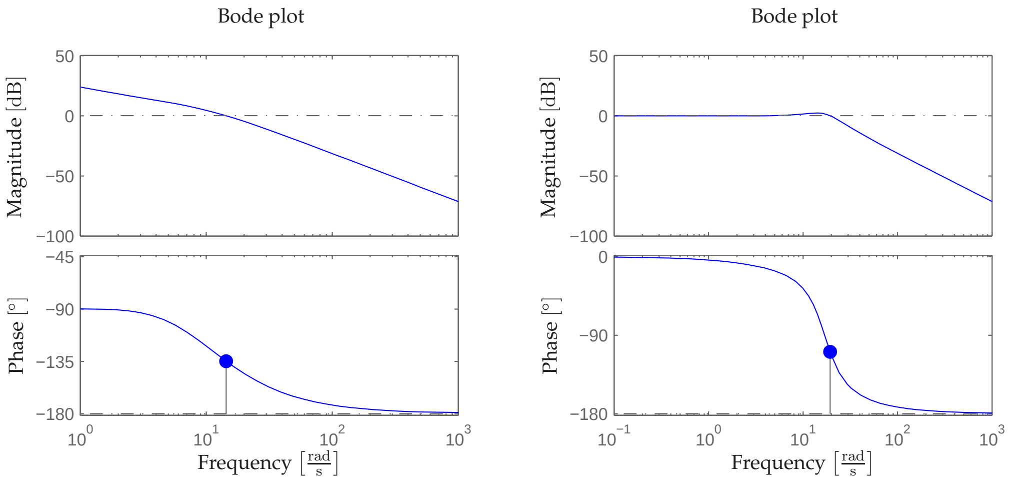
$$\Downarrow$$

$$\mathcal{C}_{{}^e\theta} = -2.54\frac{s+4}{s+10} \ , \tag{5.6}$$

where the open loop transfer function for the controlled system then becomes

$$\mathcal{T}_{^e\theta,ol} = 2.54\frac{42.51\mathrm{s} + 170}{\mathrm{s}(\mathrm{s}^2 + 17\mathrm{s} + 70)} \quad . \tag{5.7}$$

To determine whether the designed controller stabilizes the system or not is done by observering the bode plot of the open loop transfer function (see Figure 5.4(a)). From the bode plot it can be seen that the open loop system has a phase margin of approximately $45°$ and an infinite gain margin, which indicates that the system is stable in closed loop. The closed loop bode plot shows that the DC gain is equal to 1 ($0\,\mathrm{dB}$) (see Figure 5.4(b)).



(a) (b)

FIGURE 5.4: *(a) shows the bode plot of the open loop system $\mathcal{T}_{^e\theta,ol}$. (b) shows the bode plot of the closed loop system $\mathcal{T}_{^e\theta,cl}$.*

To determine the performance of the designed controller for the pitch angle, the step response of the closed loop system is examined for rise time, settling time and overshoot (see Figure 5.5). From the step response it can be seen that the rise time is approximately $0.1\,\mathrm{s}$ with an overshoot of about $20\,\%$, which indicates a fast controller. Furthermore, it can be seen that the 2% settling time for the response is approximately $0.5\,\mathrm{s}$.

From the above analysis of the designed pitch angle controller, it is concluded that it should be able to perform satisfying control of the pitch angle when applied to the nonlinear model. In addition, to test the performance of the controller the initial pitch angle of the UAV is applied an offset of $0.5\,\mathrm{rad}$ in addition to the hover operating point value of $-0.002\,\mathrm{rad}$. Furthermore, to make sure that cross couplings are not affecting the test all other states have been decoupled.

The controller is first tested on the nonlinear model using the real pitch angle as feedback (see Figure 5.6(a)). It is seen that the controller is able to stabilize ${}^e\theta$ in its hover operating point within $2\,\mathrm{s}$, which is assessed as satisfactory. Furthermore, the controller is tested using the estimated pitch angle as feedback in order to determine its robustness (see Figure 5.6(b)). As for the previous test the controller is able to stabilize ${}^e\theta$ within 2 seconds despite the measurement noise

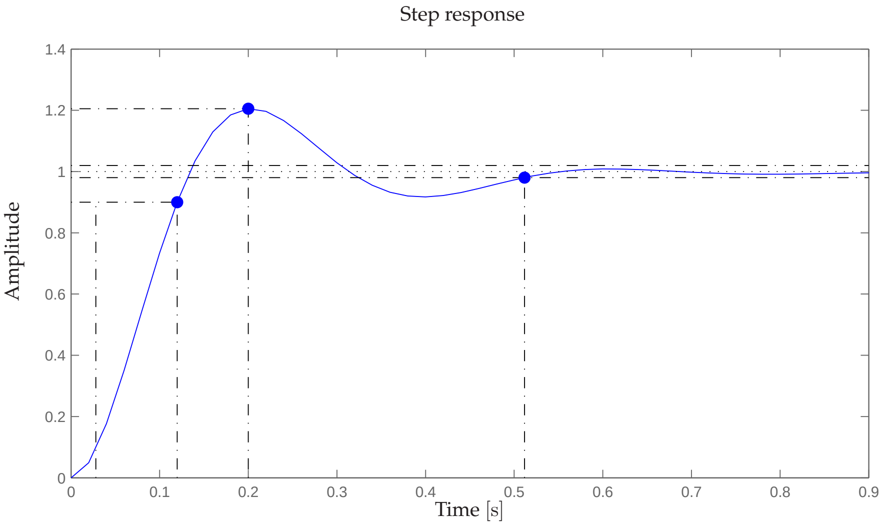


FIGURE 5.5: *Step response of the controlled system for the pitch angle $^{e}\theta$.*

in the estimated $^{e}\theta$. Based on these observations the pitch angle controller is assessed as having satisfactory performance.

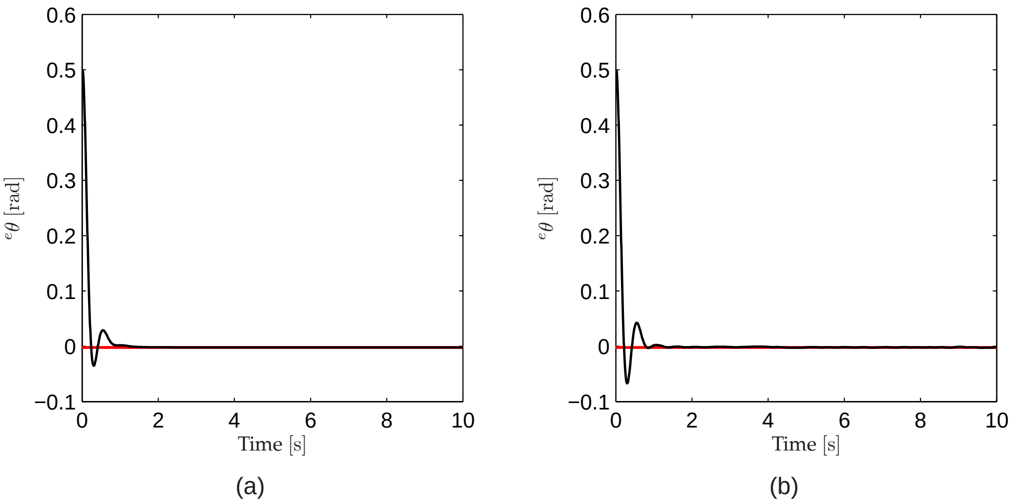


FIGURE 5.6: *Simulation result of the pitch angle controllers ability to stabilize $^{e}\theta$ using (a) model feedback and (b) estimator feedback.*

With the pitch angle controller designed it is now possible to design and test the longitudinal velocity controller. Because the longitudinal velocity controller is the outer controller in the total longitudinal controller its output must be a $^{e}\theta$ reference angle. Therefore, the transfer function can initially be identified as $G_{{}^b\dot{x}} = \frac{{}^b\dot{x}}{{}^e\theta}$. The state-space model for ${}^b\dot{x}$ can then be determined as

$$\begin{bmatrix} {}^e\dot{\theta} \\ {}^b\ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -9.82 & -0.0215 \end{bmatrix} \begin{bmatrix} {}^e\theta \\ {}^b\dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ -9.82 \end{bmatrix} {}^e\theta$$

$$\boldsymbol{y}_{{}^b\dot{x}} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} {}^e\theta \\ {}^b\dot{x} \end{bmatrix} + 0\,{}^e\theta \ .$$

From the above state-space model the transfer function $G_{{}^b\dot{x}}$ is given as (see Eq. (5.2) for calculation method)

$$G_{{}^b\dot{x}} = \frac{-9.82}{\mathrm{s} + 0.0215} \ . \tag{5.8}$$

By analyzing the root locus plot of $G_{{}^b\dot{x}}$ it can be seen that it is necessary to use a negative control gain in order to ensure stability (see Figure 5.7(a)). The compensator pole and zero are chosen such that

$$\mathcal{C}_{{}^b\dot{x}} = -K_{p,{}^b\dot{x}} \frac{\mathrm{s} + 2}{\mathrm{s} + 10} \ , \tag{5.9}$$

which result in the root locus plot depicted in Figure 5.7(b). From this root locus plot the control gain is determined as

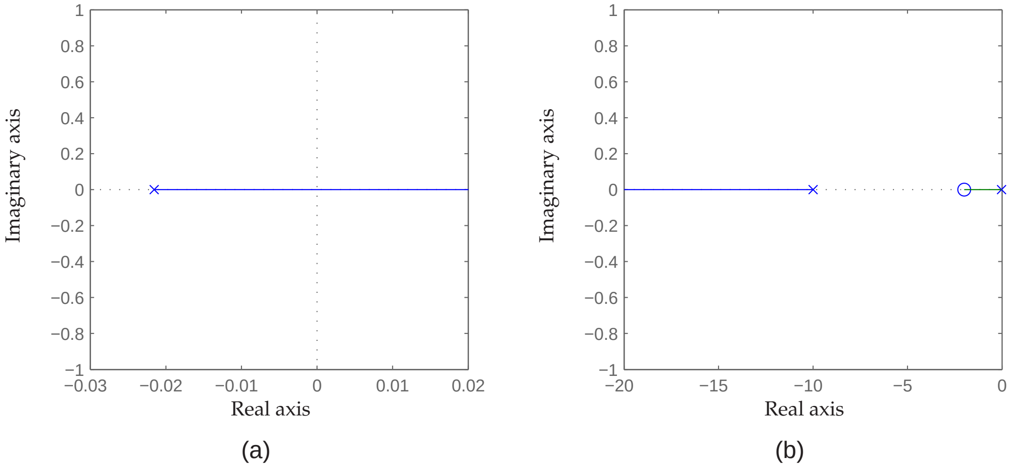


FIGURE 5.7: *(a) shows the root locus plot of $G_{{}^b\dot{x}}$. (b) shows the root locus plot of the open loop system $\mathcal{C}_{{}^b\dot{x}} G_{{}^b\dot{x}}$.*

$$K_{p,{}^b\dot{x}} = 0.14 \tag{5.10}$$

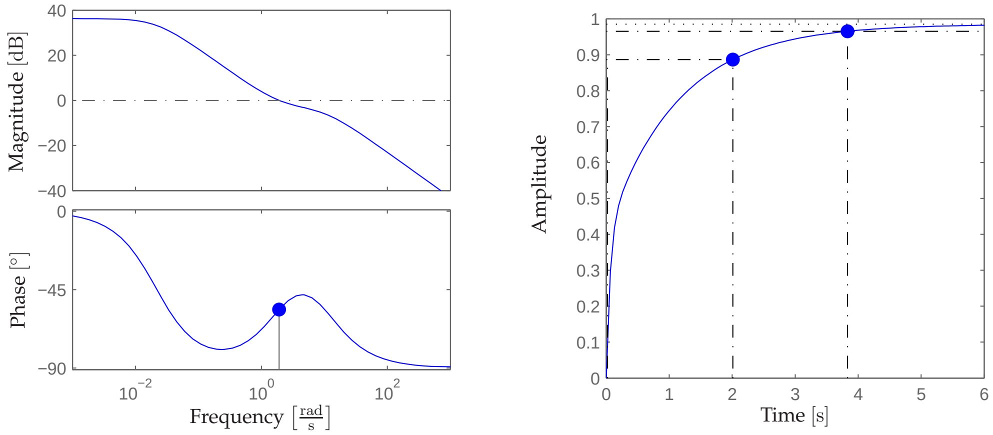
$$\Downarrow$$

$$\mathcal{C}_{{}^b\dot{x}} = -0.14 \frac{\mathrm{s} + 2}{\mathrm{s} + 10} \ , \tag{5.11}$$

which yields the open loop system

$$\mathcal{T}_{{}^b\dot{x},ol} = 0.14\frac{9.82\mathrm{s} + 19.64}{\mathrm{s}^2 + 10.02\mathrm{s} + 0.215} \quad . \tag{5.12}$$

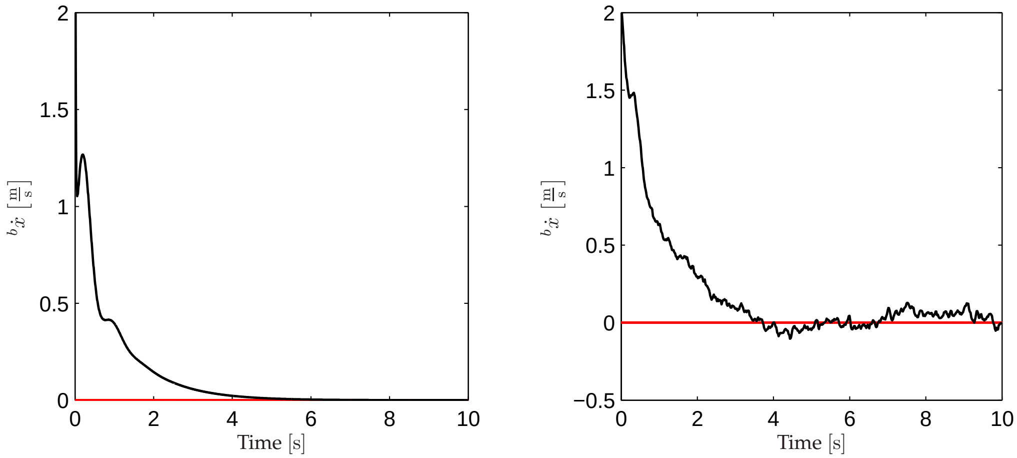
In Figure 5.8(a) the bode plot of the open loop system $\mathcal{T}_{{}^b\dot{x},ol}$ is shown, and from the bode plot it can be seen that the system has a phase margin of approximately $125\,°$ and an infinite gain margin. This indicate that the system is open loop stable and therefore also closed loop stable. In addition, the step response performance of the designed controller is analyzed (see Figure 5.8(b)). From the step response it can be seen that the rise time is $2\,\mathrm{s}$ and settling time of about $3.9\,\mathrm{s}$ indicating a stable and fast controller.



(a) (b)

FIGURE 5.8: *(a) Bode plot of the open loop system $\mathcal{T}_{{}^e\dot{x},ol}$. (b) Step response of the closed loop system $\mathcal{T}_{{}^e\dot{x},cl}$.*

To test the performance of the controller the initial longitudinal velocity of the nonlinear model is applied an offset of $2\,\frac{\mathrm{m}}{\mathrm{s}}$, where the original initial value in the hover operating point is $0\,\mathrm{m}s$. All other states then ${}^e\theta$ and ${}^b\dot{x}$ are decoupled in order to remove the effect of cross couplings. From the test simulation using the real state of ${}^b\dot{x}$ (and ${}^e\theta$ for inner controller) as feedback it is observed that the controller is able to stabilize the longitudinal velocity within 6 seconds, which is seen as being satisfactory (see Figure 5.9(a)). Furthermore, it can be seen from the test simulation using the estimated state of ${}^b\dot{x}$, that the designed controller is still able to stabilize the longitudinal velocity within 6 seconds despite the measurement noise. The designed longitudinal controller is therefore assessed usable for longitudinal velocity control.



(a) (b)

FIGURE 5.9: *Simulation result of the longitudinal velocity controllers ability to stabilize ${}^b\dot{x}$ using (a) model feedback and (b) estimator feedback.*

5.3.3 Lateral control design

The second part of the classic control design is the total lateral controller where the first step is to design the roll angle ${}^e\phi$ controller. The system matrices for ${}^e\phi$ are

$$\dot{\begin{bmatrix} {}^e\phi \\ {}^b\dot{\phi} \end{bmatrix}} = \begin{bmatrix} 0 & 1 \\ 0 & -13.86 \end{bmatrix} \begin{bmatrix} {}^e\phi \\ {}^b\dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ -85.08 \end{bmatrix} S_{\text{lat}}$$

$$\boldsymbol{y}_{{}^e\phi} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} {}^e\phi \\ {}^b\dot{\phi} \end{bmatrix} + 0\, S_{\text{lat}}.$$

From the above state-space model and (5.2) the transfer function $G_{{}^e\phi}$ is calculated as

$$G_{{}^e\phi} = \frac{-85.08}{\mathrm{s}(\mathrm{s}+13.86)} \quad . \tag{5.13}$$

From the root locus plot of $G_{{}^e\phi}$ (see Figure 5.10(a)) it can be seen that the controller must be designed with a negative control gain in order for the system to become stable. The compensator pole and zero are chosen as

$$\mathcal{C}_{{}^e\phi} = -K_{p,{}^e\phi}\frac{\mathrm{s}+5.26}{\mathrm{s}+10} \quad , \tag{5.14}$$

which yields the open loop root locus plot as depicted in Figure 5.10(b), from where the control gain is determined as

$$K_{p,{}^e\phi} \quad = \quad 3.47 \tag{5.15}$$

$$\Downarrow$$

$$\mathcal{C}_{{}^e\phi} \quad = \quad -3.47\frac{\mathrm{s}+5.26}{\mathrm{s}+10} \quad . \tag{5.16}$$

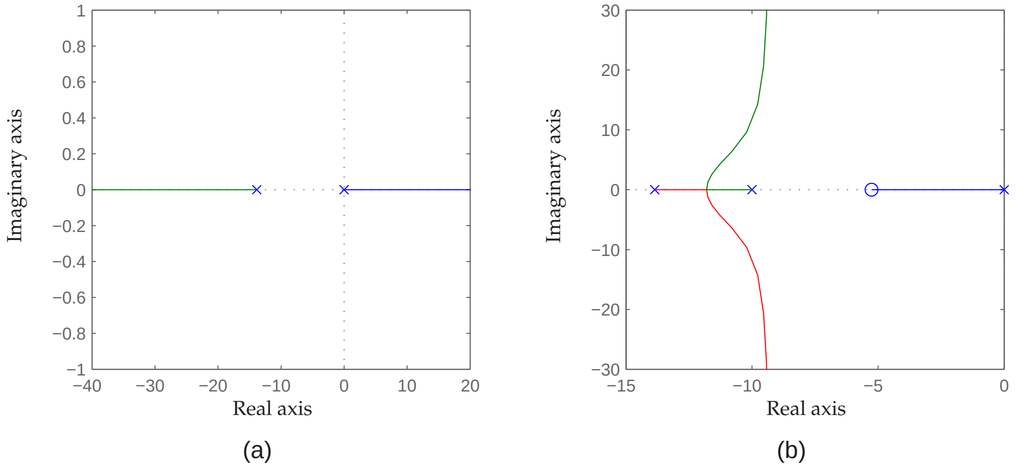


FIGURE 5.10: *(a) shows the root locus plot of $G_{^e\phi}$. (b) shows the root locus plot of the open loop system $\mathcal{C}_{^e\phi}G_{^e\phi}$.*

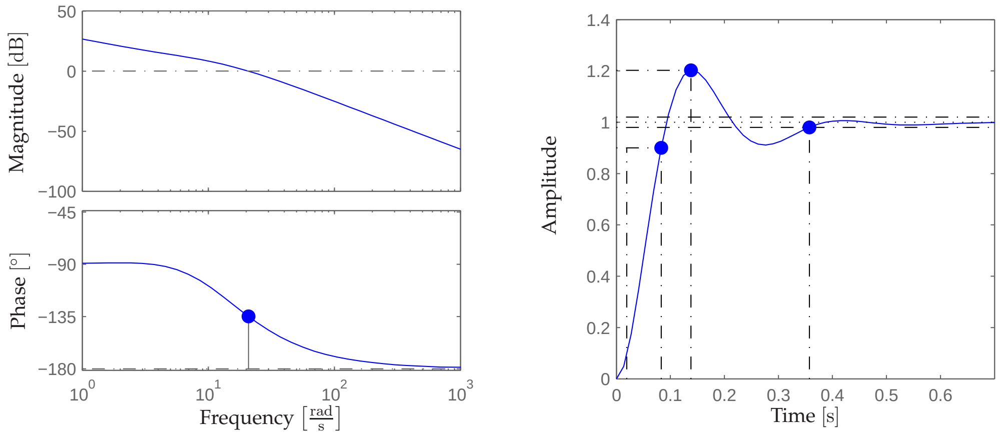
This yields the open loop system

$$\mathcal{T}_{^e\phi,ol} = 3.47\frac{16.17\mathrm{s}+85.08}{\mathrm{s}(0.1\mathrm{s}^2+2.386\mathrm{s}+13.86)} \quad . \tag{5.17}$$

From the bode plot of the open loop system $\mathcal{T}_{^e\phi,ol}$ in Figure 5.11(a), it can be seen that the system has a phase margin of $45.1°$ and an infinite gain margin, which indicate that the system is open loop stable. Furthermore, it can be seen from the step response of the closed loop system in Figure 5.11(b), that the rise time of the controlled system is approximately 0.06 seconds, and that it has an overshoot of approximately $20\,\%$. Despite the overshoot it is not considered an issue because the settling time for the controlled system is about 0.36 seconds. These observations indicate that the designed controller is able to stabilize the roll angle of the UAV.

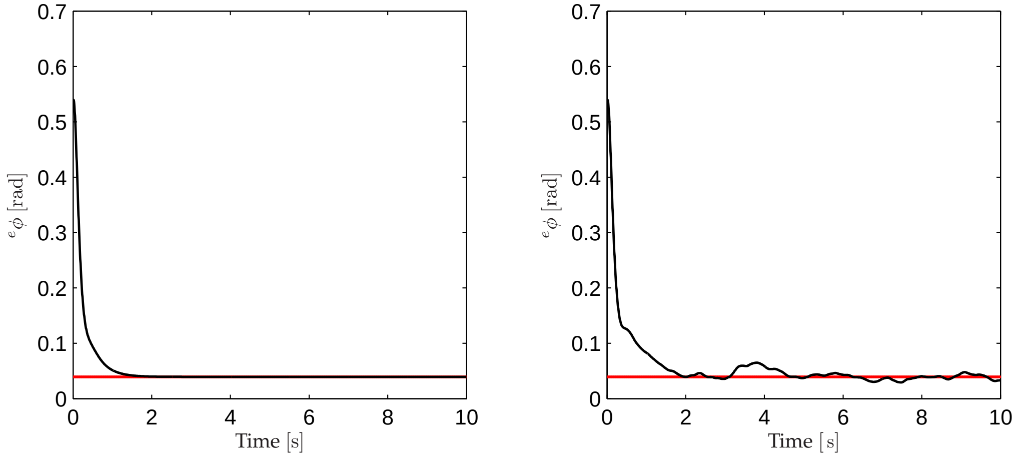
The designed controllers ability to stabilize ${}^e\phi$ is tested by applying an offset of $0.5\,\mathrm{rad}$ to the hover operating point angle value of $0.039\,\mathrm{rad}$. In addition, all other states are decoupled to remove cross couplings in the nonlinear model. In Figure 5.12(a) the controller is tested using real state of ${}^e\phi$ as feedback from which it can be seen, that the designed controller is able to stabilize the roll angle within 2 seconds. In addition, it can be seen from Figure 5.12(b) that the controller, using the estimated state of ${}^e\phi$ as feedback, is able to stabilize the roll angle within 4 seconds, which is assessed as being satisfactory.

With the roll angle controller designed it is now possible to design and test the lateral velocity controller. Because the lateral velocity controller is the outer controller in the lateral control system its output must be a ${}^e\phi$ reference angle. Therefore, the transfer function can initially be identified



(a) (b)

FIGURE 5.11: *(a) Bode plot of the open loop system $\mathcal{T}_{^e\phi,ol}$. (b) Step response of the closed loop system $\mathcal{T}_{^e\phi,cl}$.*



(a) (b)

FIGURE 5.12: *Simulation result of the roll angle controllers ability to stabilize $^e\phi$ using (a) model feedback and (b) estimator feedback.*

as $G_{^b\dot{y}} = \frac{^b\dot{y}}{^e\phi}$. The state-space model for $^b\dot{y}$ can then be identified as

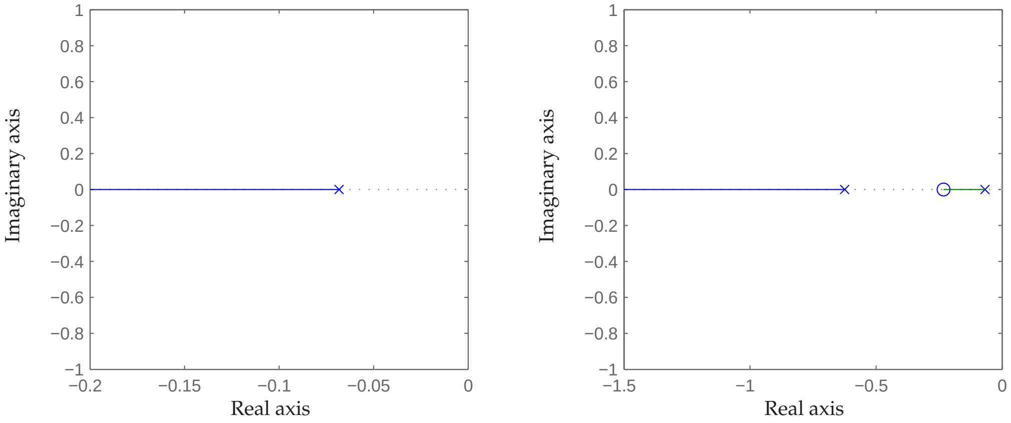
$$\dot{\begin{bmatrix} {}^e\phi \\ {}^b\dot{y} \end{bmatrix}} = \begin{bmatrix} 0 & 0 \\ 9.81 & -0.068 \end{bmatrix} \begin{bmatrix} {}^e\phi \\ {}^b\dot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ 9.81 \end{bmatrix} {}^e\phi$$

$$\boldsymbol{y}_{^b\dot{y}} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} {}^e\phi \\ {}^b\dot{y} \end{bmatrix} + 0\, {}^e\phi.$$

By using (5.2) the transfer function $G_{{}^b\dot{y}}$ is determined as

$$G_{{}^b\dot{y}} = \frac{9.81}{\mathrm{s} + 0.068} \quad . \tag{5.18}$$

By observing the root locus plot of $G_{{}^b\dot{y}}$ in Figure 5.13(a), it is seen that the system is stable and therefore must have a positive control gain. Furthermore, from the root locus plot of $G_{{}^b\dot{y}}$ the compensator pole and zero are chosen as



(a) (b)

FIGURE 5.13: *(a) shows the root locus plot of $G_{{}^b\dot{y}}$. (b) shows the root locus plot of the open loop system $\mathcal{C}_{{}^b\dot{y}}G_{{}^b\dot{y}}$.*

$$\mathcal{C}_{{}^b\dot{y}} = K_{p,{}^b\dot{y}}\frac{\mathrm{s} + 0.23}{\mathrm{s} + 0.63} \quad , \tag{5.19}$$

which result in the open loop root locus plot as seen in Figure 5.13(b) from where the controller gain is determined to be

$$K_{p,{}^b\dot{y}} = 0.56 \tag{5.20}$$

$$\Downarrow$$

$$\mathcal{C}_{{}^b\dot{y}} = 0.56\frac{\mathrm{s} + 0.23}{\mathrm{s} + 0.63} \quad . \tag{5.21}$$

The open loop transfer function then becomes

$$\mathcal{T}_{{}^b\dot{y},ol} = 0.56\frac{42.19\mathrm{s} + 9.812}{1.6\mathrm{s}^2 + 1.109\mathrm{s} + 0.068} \quad , \tag{5.22}$$

which result in the bode plot in Figure 5.14(a). From the bode plot the phase margin of the open loop system is determined to be approximately $91\,°$, and that it has an infinite gain margin, which indicate that the system is open loop stable. In addition, analyzing the step response of the closed loop system it is observed that the rise and settling time are about 0.1 and 0.4 seconds respectively (see Figure 5.14(b)). Furthermore, it is noticed that the controlled system is over damped.

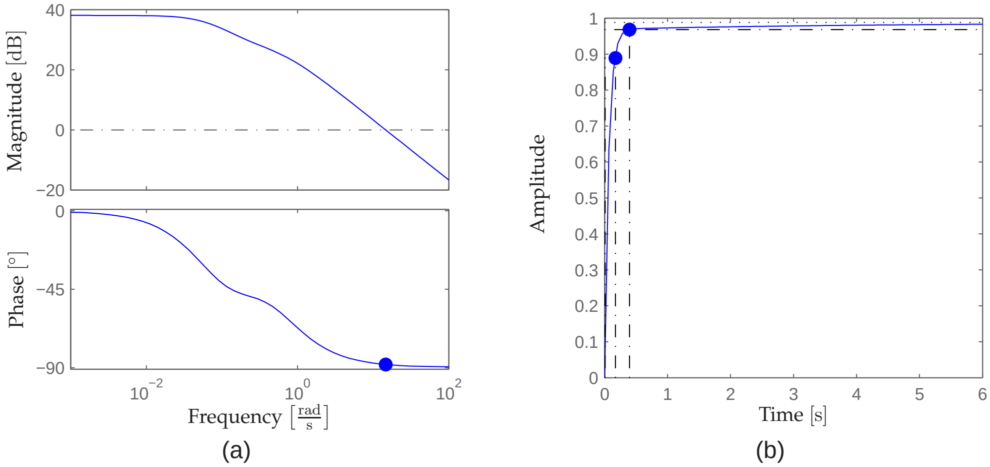


FIGURE 5.14: *(a) Bode plot of the open loop system $\mathcal{T}_{^b\dot{y},ol}$. (b) Step response of the closed loop system $\mathcal{T}_{^b\dot{y},cl}$.*

However, the fact that the controlled system is over damped is not considered an issue because of the small rise and settling times. Based on these observations it is concluded that the designed controller should be able to stabilize the lateral velocity of the UAV.

The controller is tested by applying an offset to the initial lateral velocity of $2\,\frac{\mathrm{m}}{\mathrm{s}}$, where the hover operating point value is $0\,\frac{\mathrm{m}}{\mathrm{s}}$. In addition, all other states than ${}^e\phi$ and ${}^b\dot{y}$ have been decoupled to remove cross couplings. Using the real state of ${}^b\dot{y}$ as feedback, it can be seen from Figure 5.15(a) that the designed controller is able to stabilize the lateral velocity within 15 seconds, which is rather long compared to previous settling times. However, it is deemed usable. Furthermore, it can be seen from Figure 5.15(b), that the designed controller, using the estimated state of ${}^b\dot{y}$ as feedback, is able to stabilize the lateral velocity within 6 seconds with the rather noisy signal taken into consideration. Based on these observations the lateral velocity controller is assessed usable.

5.3.4 Vertical control design

The third part of classic control design is the vertical velocity control, and its state-space system is determined as

$$\begin{bmatrix} {}^e\dot{z} \\ {}^b\ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 0.9992 \\ 0 & -0.9001 \end{bmatrix} \begin{bmatrix} {}^e z \\ {}^b\dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ -98.5353 \end{bmatrix} S_{\mathrm{col}}$$

$$\boldsymbol{y}_{{}^b\dot{z}} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} {}^e z \\ {}^b\dot{z} \end{bmatrix} + 0\,S_{\mathrm{col}}.$$

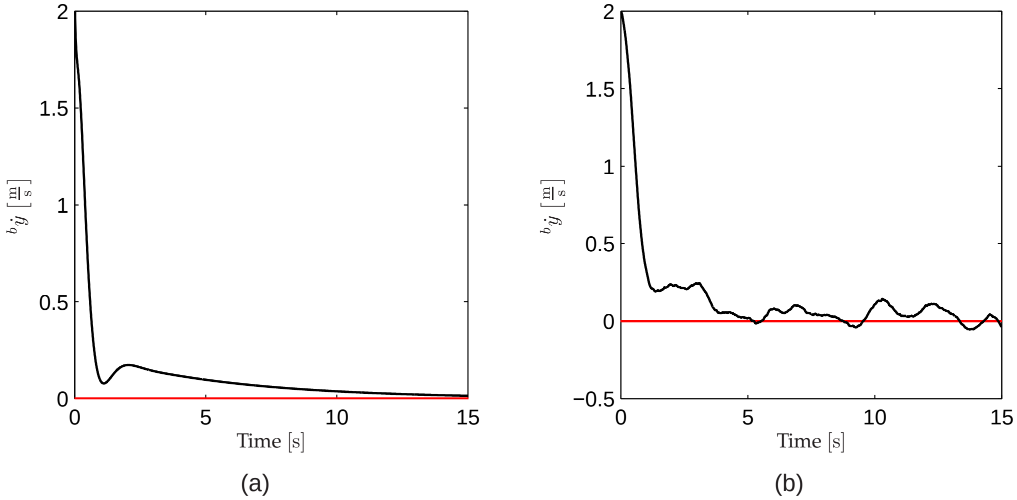


FIGURE 5.15: *Simulation result of the lateral velocity controllers ability to stabilize $^b\dot{y}$ using (a) model feedback and (b) estimator feedback.*

Based on the above state-space system the transfer function $G_{^b\dot{z}} = \frac{^b\dot{z}}{S_{\text{col}}}$ is calculated as

$$G_{^b\dot{z}} = \frac{-98.54}{s + 0.9} \quad , \tag{5.23}$$

which result in the root locus plot as depicted in Figure 5.16(a). From the root locus plot it can be determined that a negative control gain is necessary to stabilize the system. Based on the root

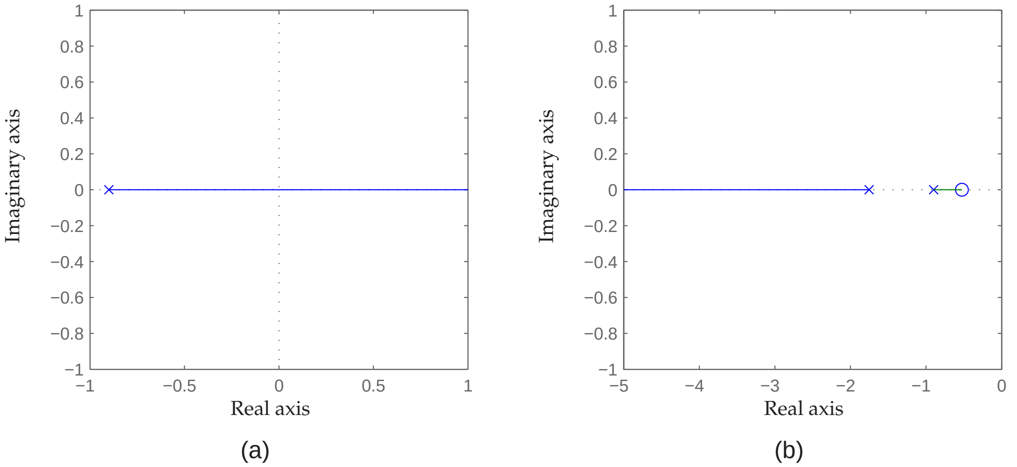


FIGURE 5.16: *(a) shows the root locus plot of $G_{^b\dot{z}}$. (b) shows the root locus plot of the open loop system $C_{^b\dot{z}}G_{^b\dot{z}}$.*

locus plot of $G_{^b\dot{z}}$ the compensator pole and zero are determined as

$$\mathcal{C}_{^b\dot{z}} = -K_{p,^b\dot{z}} \frac{\mathrm{s}+0.53}{\mathrm{s}+1.75} \quad , \tag{5.24}$$

which yields the open loop root locus plot in Figure 5.16(b). The control gain is then determined as

$$K_{p,^b\dot{z}} = 0.17 \tag{5.25}$$

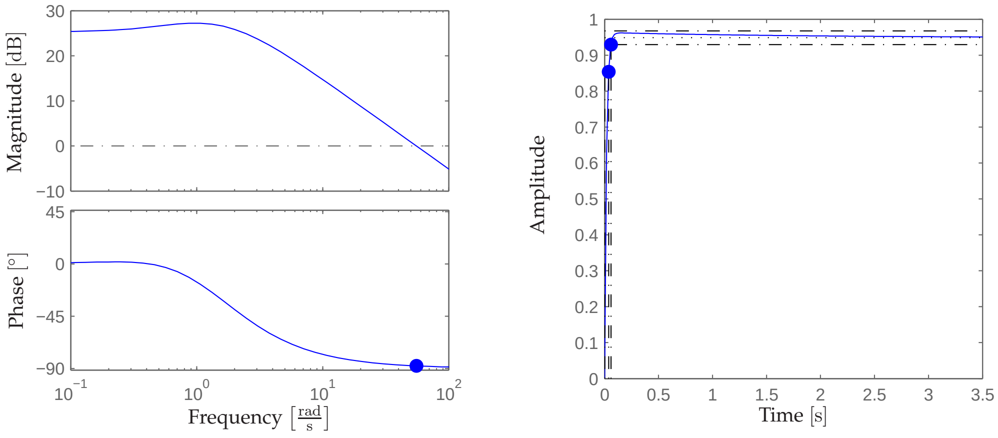
$$\Downarrow$$

$$\mathcal{C}_{^b\dot{z}} = -0.17 \frac{\mathrm{s}+0.53}{\mathrm{s}+1.75} \quad , \tag{5.26}$$

which then yields the open loop transfer function as

$$\mathcal{T}_{^b\dot{z},ol} = 0.169 \frac{187.2\mathrm{s}+98.54}{0.57\mathrm{s}^2+1.513\mathrm{s}+0.9} \quad . \tag{5.27}$$

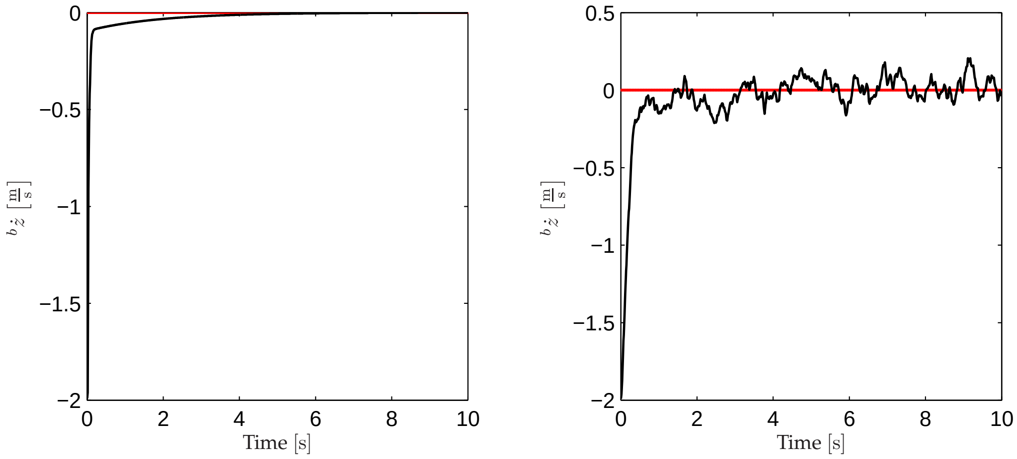
From the bode plot of the open loop system $\mathcal{T}_{^b\dot{z},ol}$ in Figure 5.17(a) it is observed that the system has a phase margin is approximately $92\,°$ and an infinite gain margin, which indicate that the system is open loop stable and therefore also closed loop stable. Analyzing the step response of the closed loop system $\mathcal{T}_{^b\dot{z},cl}$ (see Figure 5.17(b)) it can be seen that the controlled system has a rise time of about 0.03 seconds, a settling time of about $0.06\,\mathrm{s}$ and no overshoot, which indicate a fast controller. Therefore, it can be concluded that the controller should be able to stabilize the vertical velocity.



(a) (b)

FIGURE 5.17: *(a) Bode plot of the open loop system $\mathcal{T}_{^b\dot{z},ol}$. (b) Step response of the closed loop system $\mathcal{T}_{^b\dot{z},cl}$.*

The controller is tested by applying an offset to the initial vertical velocity of $-2\,\frac{\mathrm{m}}{\mathrm{s}}$, where the hover operating point is $0\,\frac{\mathrm{m}}{\mathrm{s}}$. In addition, all other states have been decoupled to remove cross couplings. Using the real state of $^b\dot{z}$ as feedback it can be observed that the controller is able to



(a) (b)

FIGURE 5.18: *Simulation result of the vertical velocity controllers ability to stabilize ${}^{b}\dot{z}$ using (a) model feedback and (b) estimator feedback.*

stabilize the vertical velocity within 4 seconds (see Figure 5.18(a)). Furthermore, the controller is tested using the estimated state of ${}^{b}\dot{z}$ as feedback, where it can be determined that the controller is also able to stabilize the vertical velocity here within 4 seconds despite measurement noise (see Figure 5.18(b)). Based on the observations done the vertical velocity controller will be used.

5.3.5 Yaw control design

The fourth and last par of the classic control design is for the yaw angle and its state-space system is determined as

$$\begin{bmatrix} {}^{e}\dot{\psi} \\ {}^{b}\ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0.9992 \\ 0 & -4.4331 \end{bmatrix} \begin{bmatrix} {}^{e}\psi \\ {}^{b}\dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ 3.6196 \end{bmatrix} S_{\text{tr}}$$

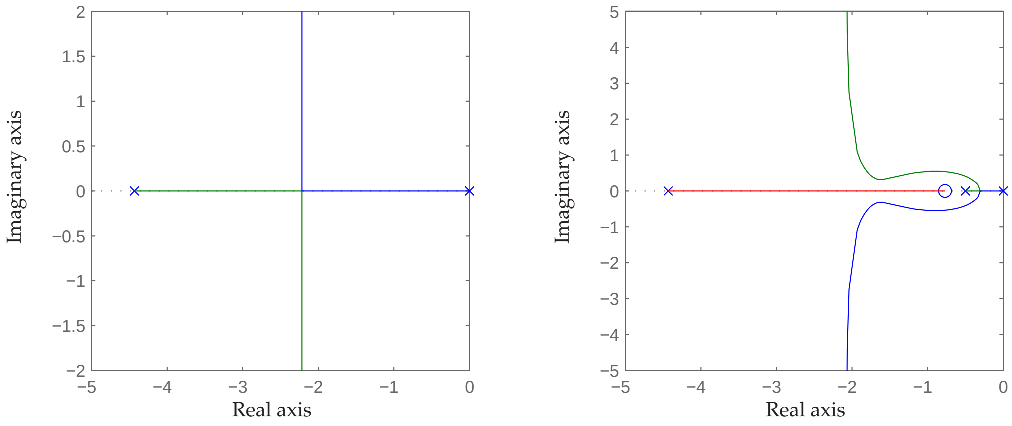
$$\boldsymbol{y}_{{}^{e}\psi} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} {}^{e}\psi \\ {}^{b}\dot{\psi} \end{bmatrix} + 0\, S_{\text{tr}}.$$

From the state-space system the transfer function $G_{{}^{e}\psi} = \frac{{}^{e}\psi}{S_{\text{tr}}}$ is determined as

$$G_{{}^{e}\psi} = \frac{3.617}{\text{s}(\text{s} + 4.433)} \quad , \tag{5.28}$$

which result in the root locus plot seen in Figure 5.19(a). From the root locus plot of $G_{{}^{e}\psi}$ it is determined that a positive control gain is necessary. In addition, based on the root locus plot the compensator pole and zero are chosen as

$$\mathcal{C}_{{}^{e}\psi} = K_{p,{}^{e}\psi} \frac{\text{s} + 0.77}{\text{s} + 0.5} \quad , \tag{5.29}$$



(a) (b)

FIGURE 5.19: *(a) shows the root locus plot of $G_{^e\psi}$. (b) shows the root locus plot of the open loop system $\mathcal{C}_{^e\psi}G_{^e\psi}$.*

which result in the open loop root locus plot seen in Figure 5.19(b). Based on the root locus plot of the open loop system $\mathcal{C}_{^e\psi}G_{^e\psi}$ the control gain is determined as

$$K_{p,{^e\psi}} = 2.53 \tag{5.30}$$

$$\Downarrow$$

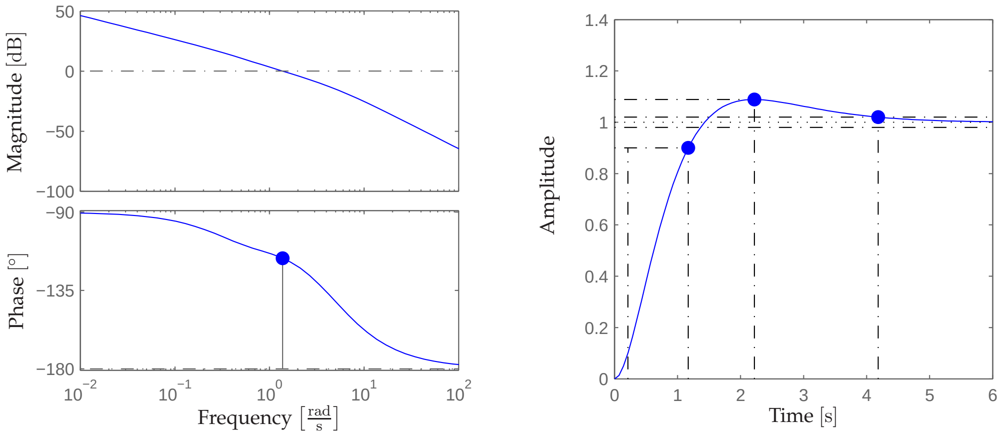
$$\mathcal{C}_{^e\psi} = 2.53\frac{\mathrm{s}+0.77}{\mathrm{s}+0.5} \quad , \tag{5.31}$$

which result in the open loop transfer function

$$\mathcal{T}_{^e\psi,ol} = 2.53\frac{4.702\mathrm{s}+3.617}{\mathrm{s}(2\mathrm{s}^2+9.866s+4.433)} \quad . \tag{5.32}$$

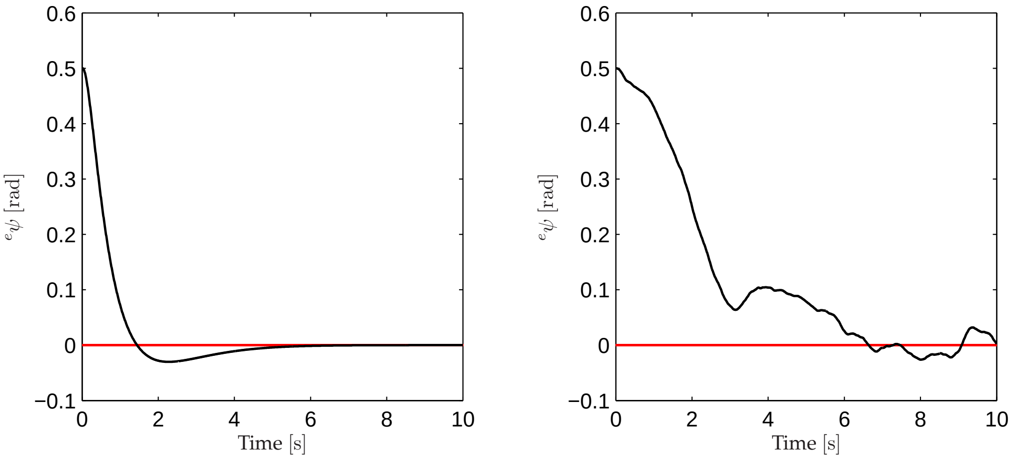
From the bode plot of the open loop system $\mathcal{T}_{^e\psi,ol}$ it can be observed, that the system has a phase margin of approximately $65\,°$ and an infinite gain margin, which indicate that the system is open loop stable (see Figure 5.20(a)). Furthermore, from the step response of the closed loop system in Figure 5.20(b), it is seen that the controlled system has a rise time of about 1 second and an overshoot of approximately $9\,\%$, but this overshoot is not considered an issue since it is rather small, and that the settling time of the controlled system is about 4.2 seconds. This show that the designed controller shoould be able to stabilize the yaw angle.

The yaw angle controller is tested by applying an offset of $0.5\,\mathrm{rad}$ to the initial yaw angle of $0\,\mathrm{rad}$. Furthermore, all other states are decoupled to remove cross couplings. From Figure 5.21(a) it can be observed that the controller is able to stabilize the yaw angle using the real state of ${}^e\psi$ within 6 seconds. In addition, it can be observed from Figure 5.21(b) that the controller also is able to stabilize the yaw angle when using the estimated state of ${}^e\psi$ as feedback although within 8 seconds. It is therefore assessed that the yaw angle controller can be used.



(a) (b)

FIGURE 5.20: *(a) Bode plot of the open loop system $\mathcal{T}_{^e\psi,ol}$. (b) Step response of the closed loop system $\mathcal{T}_{^e\psi,cl}$.*



(a) (b)

FIGURE 5.21: *Simulation result of the yaw angle controllers ability to stabilize $^e\psi$ using (a) model feedback and (b) estimator feedback.*

Having designed all of the necessary SISO controllers to maintain the UAV in the hover operating point a more powerful stress test of the total control system will be performed.

5.4 Classic control system test

In the following the total classic control system will be tested for its stationary and dynamical properties. The individual designed controllers from the previous section are implemented in the control structure depicted in Figure 5.1 on page 26.

5.4.1 Stationary properties

The total control system will be tested for its ability to maintain the UAV in the hover operating point, when the nonlinear model is initialized in this point. Note that early tests showed some of the controllers had too high control gains, why it is necessary to lower these gains by applying additional scalars to the controllers. Having modified control system the aforementioned test is performed, where the simulation results can be seen in Figure 5.22. From the plotted simulation data it can be observed that the roll and pitch angles exhibit strong oscillating behavior about their respective operating point values, which indicate fast transient responses of the two con-

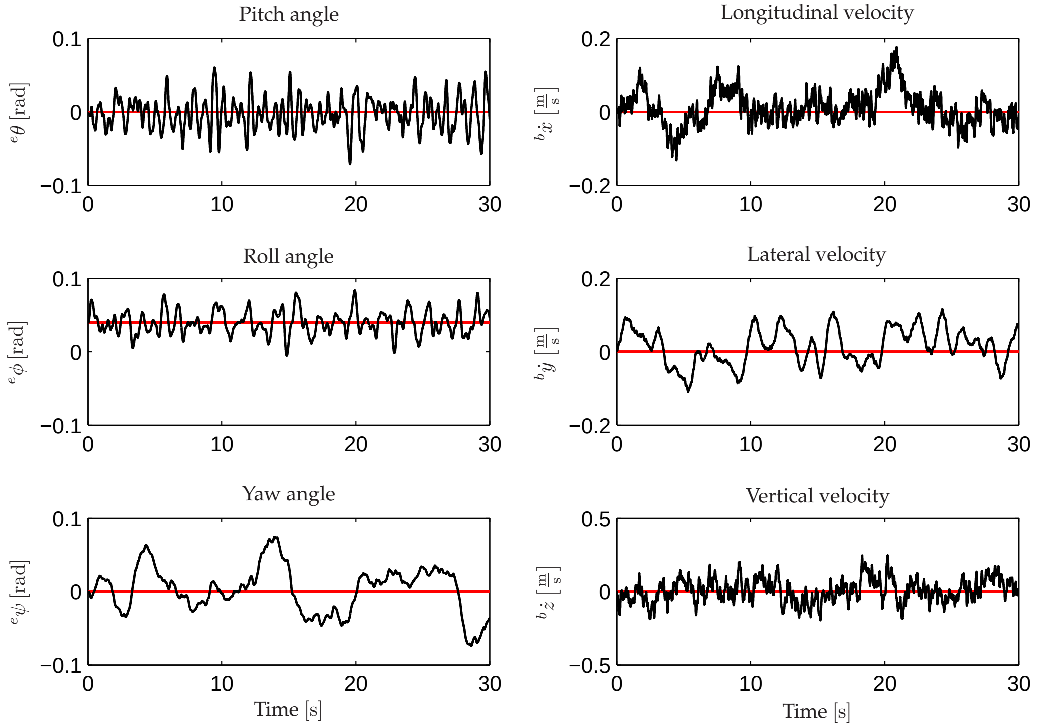


FIGURE 5.22: *Simulation result of the total classic control systems ability to maintain the UAV in the hover operating point. Strong oscillating behavior can be observed in some of the controlled states, which indicate fast transient responses of their respective controllers.*

trollers. In contrast to the roll and pitch angles, the yaw angle show more slow behavior with respect to the estimator feedback signal, which ultimately result in slow tracking of the reference signal. Common for the three angles is that they vary less than $0.1\,\mathrm{rad}$ from their respective operating point values. Observations of the longitudinal and vertical velocities also exhibit oscillating behavior, which, like for the roll and pitch angles, indicate fast transient responses of the two respective controllers. The lateral velocity also show signs of oscillating behavior, however, not as noticeable as for the other velocities. All of the velocities follow their respective operating point values with a maximum variation of $0.3\,\frac{\mathrm{m}}{\mathrm{s}}$.

Based on the above observations regarding the control systems ability to maintain the UAV in the hover operating point, it is concluded that the designed classic linear control system is capable of maintaining the UAV in hover.

5.4.2 Dynamical proporties

The dynamical test is performed in the exact same manner as each individual test of the different controllers. However, here all cross couplings are not removed and all controlled outputs are given an initial offset corresponding to the offsets used for each individual controller test (see Table 4.1 on page 22 for an overview of the offsets). The test results can be seen in Figure 5.23.

To be able to determine when the UAV can be identified as being in hover the variation values of $0.1\,\mathrm{rad}$ and $0.3\,\frac{\mathrm{m}}{\mathrm{s}}$, for the angles and velocities respectively, is used as limits. From Figure 5.23 it can then be observed that all controlled states have settled within a period of 5 seconds except the lateral velocity, which settles after approximately 10 seconds. It is based on these observations concluded that the designed classic linear control system is capable of stabilizing the UAV in the hover operating point and capable of rejecting possible disturbances.

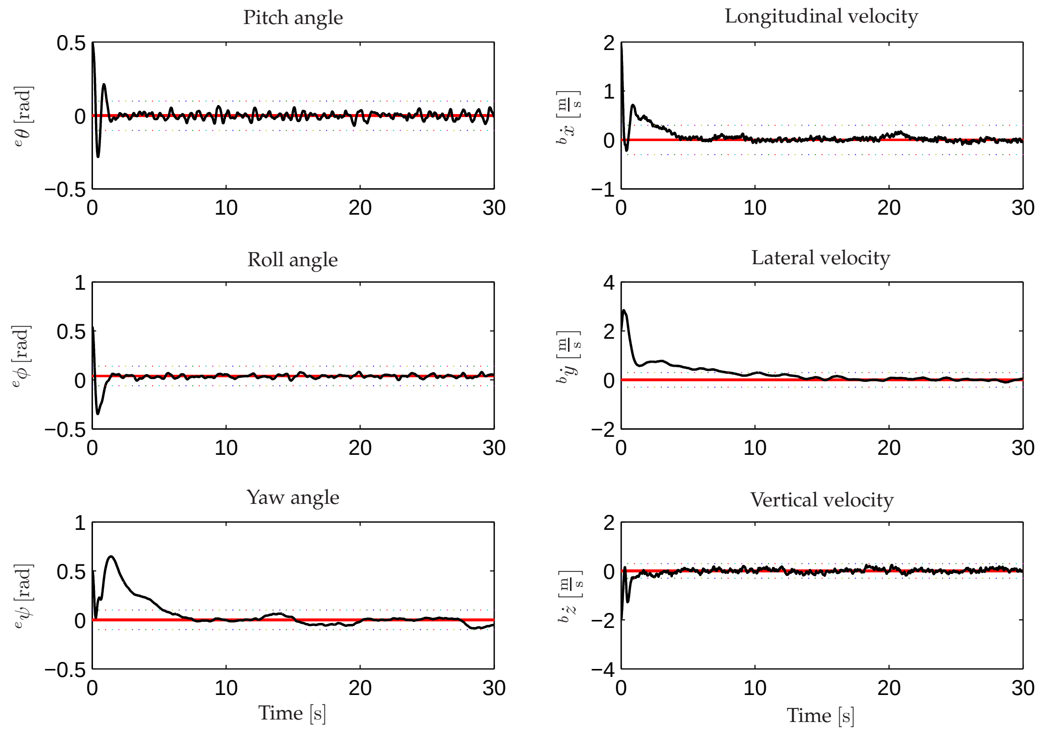


FIGURE 5.23: *Simulation result of the total classic control systems ability to stabilize the UAV in the hover operating point when given an initial offset in all of the controlled states. The total settling time for the controlled system is prolonged due to slow lateral velocity control.*

6

Optimal linear quadratic control

This chapter describes the design and implementation of the LQR hover and forward flight controllers. Since the LQR method is a MIMO controller it results in design of only two different controllers; one for hover and one for forward flight. Before the actual controller design is performed the choice of operating points and control states are described, followed by the principle of the LQR method.

6.1 Choice of operating points and feedback states

As the first step in designing the hover and forward flight controllers the two operating points for linearization of the nonlinear model must be determined, which is done in the following.

6.1.1 Choice of operating points

As with the classic hover controller the LQR controller design is based on linearization of the nonlinear model in a chosen operating point. As described in the introduction to this thesis two controllers, one for hover and one for forward flight, must be developed using the LQR method for later use in gain scheduling. Consequently two operating points must be determined prior to the design. The first operating point for the hover controller is obviously the hover condition itself.

The second operating point for the forward flight controller is, however, not given in advance. As stated in the introduction it is desired to complete level 1 in the IARC as fast as possible. An obvious choice of forward flight operating point is therefore the maximum speed of the UAV, which is known to be at least $92\,\frac{\mathrm{km}}{\mathrm{h}}$. However, as the use of gain scheduling provides for the use of several forward flight controllers for different forward flight speeds, the second operating point is instead chosen to be at $10\,\frac{\mathrm{m}}{\mathrm{s}}$ to allow the for possibility of implementing several forward flight controllers, where this thesis focuses on the usage of two controllers.

6.1.2 Choice of control, reference and integral states

As described in Section 3.2 the feedback states for the linear controller are given as in (6.1). Hence, the supervisor must handle the position control of the control system.

$$\boldsymbol{x}_s = \left[\begin{array}{ccccccccc} {}^e\phi & {}^e\theta & {}^e\psi & {}^b\dot{x} & {}^b\dot{y} & {}^b\dot{z} & {}^b\dot{\phi} & {}^b\dot{\theta} & {}^b\dot{\psi} \end{array} \right]^T \tag{6.1}$$

The index s (representing the system feedback states) is used for the sake of clarity throughout this chapter, as state augmentation in subsequent sections involves more indices. At a first sight the translatory velocity states $[{}^b\dot{x}\ \ {}^b\dot{y}\ \ {}^b\dot{z}]$ together with the heading ${}^e\psi$ seem sufficient for controlling the UAV through a given level 1 specification. However, as the LQR control method utilizes weights to determine the influence of states, it is chosen to design the controller for all nine states in (6.1). In this way the largest possible freedom regarding choice of weights is achieved.

As described in Section 4.1 the LQR controllers must feature both reference and integral states, which are included in the LQR controller by state augmentation. As with the system states the reference and integral states are being weighted, therefore it is chosen to include reference and integral states for all nine states in (6.1) as well.

In the next section the LQR principle is introduced, and the state augmentation for including reference and integral states is explained.

6.2 The principle of LQR

The LQR control design done in this thesis is based on the procedures in Sørensen [1992], which are described in the following. It relies on minimization of the performance index with weight matrices on states and inputs. Furthermore, it features both reference and integral states, which are needed in this thesis as described in the previous section.

The LQR control method is based on the state equation from the general discrete state-space description of the system:

$$\boldsymbol{x}_s(k+1) = \boldsymbol{\Phi}_s\boldsymbol{x}_s(k) + \boldsymbol{\Gamma}_s\boldsymbol{u}(k) \tag{6.2}$$

$$\boldsymbol{y}_s(k) = \boldsymbol{H}_s\boldsymbol{x}_s(k) + \boldsymbol{P}_s\boldsymbol{u}(k)\ \ , \tag{6.3}$$

where $\boldsymbol{x}_s(k)$ and $\boldsymbol{u}(k)$ are given by (6.1) and (6.4) respectively.

$$\boldsymbol{u} = \begin{bmatrix} S_{\text{col}} & S_{\text{lat}} & S_{\text{lon}} & S_{\text{tr}} \end{bmatrix}^T \tag{6.4}$$

The LQR principal builds on determining the optimal input $\boldsymbol{u}^*(k)$ such that the performance index given in (6.5) is minimized.

$$\mathcal{I} = \sum_{k=0}^{N-1}\Big(\boldsymbol{x}_s^T(k)\boldsymbol{Q}_1\boldsymbol{x}_s(k) + \boldsymbol{u}^T(k)\boldsymbol{Q}_2\boldsymbol{u}(k)\Big) + \boldsymbol{x}_s^T(N)\boldsymbol{Q}_N\boldsymbol{x}_s(N) \tag{6.5}$$

The quadratic weight matrices $\boldsymbol{Q}_1$, $\boldsymbol{Q}_2$ and $\boldsymbol{Q}_N$ weighting the states $\boldsymbol{x}_s(k)$, the inputs $\boldsymbol{u}(k)$ and the final states $\boldsymbol{x}_s(N)$ respectively. These are all chosen along with the time horizon N, therefore the determination of the weight matrices are the actual design task in developing an LQR controller. Because the weight matrices are chosen it results in a trade-off between achieving small errors in the control states and small control signals.

The optimal input $\boldsymbol{u}^*(k)$ can be found by the control law

$$\boldsymbol{u}^*(k) = -\boldsymbol{L}(k)\boldsymbol{x}_s(k)\ \ . \tag{6.6}$$

The proportional matrix $\boldsymbol{L}(k)$ in (6.6) is given as:

$$\boldsymbol{L}(k) = \left[\boldsymbol{Q}_2 + \boldsymbol{\Gamma}_s^T \boldsymbol{S}(k+1)\boldsymbol{\Gamma}_s\right]^{-1} \boldsymbol{\Gamma}_s^T \boldsymbol{S}(k+1)\boldsymbol{\Phi}_s \ , \tag{6.7}$$

where $\boldsymbol{S}(k)$ satisfies the following discrete time dynamic Ricatti equation:

$$\boldsymbol{S}(k) = \boldsymbol{Q}_1 + \boldsymbol{\Phi}_s^T \boldsymbol{S}(k+1)[\boldsymbol{\Phi}_s - \boldsymbol{\Gamma}_s \boldsymbol{L}(k)] \ . \tag{6.8}$$

It is desired to have a constant feedback gain $\boldsymbol{L}(0)$ calculated in advance in a closed loop control system as shown in Figure 6.1. This is obtained by evaluating (6.7) and (6.8) backwards in time, with $\boldsymbol{S}(k) = \boldsymbol{Q}_N$ and $k = N-1$ initially, until $k = 0$. In this way the performance index is at all times kept N steps ahead in time, where $k = 0$ denotes the current time. Prior to the calculation of the controller $\boldsymbol{L}(0)$, the weight matrices $\boldsymbol{Q}_1$, $\boldsymbol{Q}_2$ and $\boldsymbol{Q}_N$ and the time horizon N must be chosen.

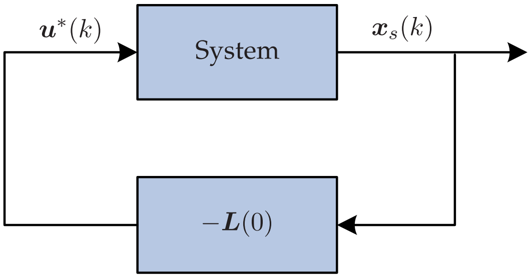


FIGURE 6.1: *The structure of the closed loop system for an LQR controller.*

6.2.1 Design parameters

The system having nine states and four inputs yields the following dimensions of the weight matrices $\boldsymbol{Q}_1 \in \mathbb{R}^{9\times 9}$ and $\boldsymbol{Q}_2 \in \mathbb{R}^{4\times 4}$. To simplify the design process it is chosen to use diagonal weight matrices, which results in a total of 13 weights to be determined. In doing so, each diagonal entry in $\boldsymbol{Q}_1(j,j)$ and $\boldsymbol{Q}_2(j,j)$ is interpreted as a measure of the relative weights of $x_j(k)$ and $u_j(k)$ respectively. A commonly used method for choosing the entries, is based on a physical insight in the system. If the open loop system is subject to limitations on inputs and states, the diagonal entries can as a starting point be chosen as:

$$\boldsymbol{Q}_1(j,j) = \frac{1}{x_{j,\max}^2} \qquad \boldsymbol{Q}_2(j,j) = \frac{1}{u_{j,\max}^2} \ . \tag{6.9}$$

In addition, a weight factor can be introduced to weight $\boldsymbol{Q}_1$ relative to $\boldsymbol{Q}_2$. The weight on the final states are chosen as $\boldsymbol{Q}_N = \boldsymbol{Q}_1$, which is often the case when the value of the final states are of no particular importance (see Sørensen [1992]).

The size of the time horizon N is determined by trial-and-error, by keeping in mind that a large value of N improves the stationary properties of the control system, and that a small value of N improves the dynamical properties of the control system (see Sørensen [1992]).

6.2.2 Implementation of reference states

The reference states are implemented in the control design such that the controller is able to follow desired set-points. The set-points are provided by the supervisor and therefore modeled as constants, yielding the following state-space representation of the system together with the reference:

$$\boldsymbol{x}_s(k+1) = \boldsymbol{\Phi}_s \boldsymbol{x}_s(k) + \boldsymbol{\Gamma}_s \boldsymbol{u}(k) \tag{6.10}$$

$$\boldsymbol{y}_s(k) = \boldsymbol{H}_s \boldsymbol{x}_s(k) + \boldsymbol{P}_s \boldsymbol{u}(k) \tag{6.11}$$

$$\boldsymbol{x}_r(k+1) = \boldsymbol{\Phi}_r \boldsymbol{x}_r(k) \tag{6.12}$$

$$\boldsymbol{r}(k) = \boldsymbol{H}_r \boldsymbol{x}_r(k) \ \ , \tag{6.13}$$

where $\boldsymbol{\Phi}_r$ and $\boldsymbol{H}_r$ are 9×9 identity matrices. With the control error given as

$$\boldsymbol{e}(k) = \boldsymbol{r}(k) - \boldsymbol{y}_s(k) = \boldsymbol{H}_r \boldsymbol{x}_r(k) - \boldsymbol{H}_s \boldsymbol{x}_s(k) \ \ , \tag{6.14}$$

the performance index becomes:

$$\mathcal{I} = \sum_{k=0}^{N-1} \left(\boldsymbol{e}^T(k) \boldsymbol{Q}_{1e} \boldsymbol{e}(k) + \boldsymbol{u}^T(k) \boldsymbol{Q}_2 \boldsymbol{u}(k) \right) + \boldsymbol{e}^T(N) \boldsymbol{Q}_{Ne} \boldsymbol{e}(N) \ \ , \tag{6.15}$$

where $\boldsymbol{Q}_{1e}$ is the control error weight matrix. To calculate the gain matrices for the states and references, the state vector is augmented to contain both system and reference states:

$$\boldsymbol{x}_{\text{aug}}(k) = \begin{bmatrix} \boldsymbol{x}_s^T(k) & \boldsymbol{x}_r^T(k) \end{bmatrix}^T \ \ . \tag{6.16}$$

The state-space representation for the augmented state then becomes

$$\begin{bmatrix} \boldsymbol{x}_s(k+1) \\ \boldsymbol{x}_r(k+1) \end{bmatrix} = \left[\begin{array}{c|c} \boldsymbol{\Phi}_s & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{\Phi}_r \end{array} \right] \begin{bmatrix} \boldsymbol{x}_s(k) \\ \boldsymbol{x}_r(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Gamma}_s \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{u}(k) \tag{6.17}$$

$$\boldsymbol{e}(k) = \begin{bmatrix} -\boldsymbol{H}_s & \boldsymbol{H}_r \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_s(k) \\ \boldsymbol{x}_r(k) \end{bmatrix} \ \ . \tag{6.18}$$

The augmented system can now be described as:

$$\boldsymbol{x}_{\text{aug}}(k+1) = \boldsymbol{\Phi}_{\text{aug}} \boldsymbol{x}_{\text{aug}}(k) + \boldsymbol{\Gamma}_{\text{aug}} \boldsymbol{u}(k) \tag{6.19}$$

$$\boldsymbol{e}(k) = \boldsymbol{H}_{\text{aug}} \boldsymbol{x}_{\text{aug}}(k) \ \ . \tag{6.20}$$

The augmented system can, as shown for the system in (6.2) and (6.3), be used to calculate the system and reference feedback gains depicted in Figure 6.2 for the optimal controller:

$$\boldsymbol{u}(k) = -\boldsymbol{L}_{\text{aug}}(0) \boldsymbol{x}_{\text{aug}}(k) = -\begin{bmatrix} \boldsymbol{L}(0) & \boldsymbol{L}_r(0) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_s(k) \\ \boldsymbol{x}_r(k) \end{bmatrix} \ \ . \tag{6.21}$$

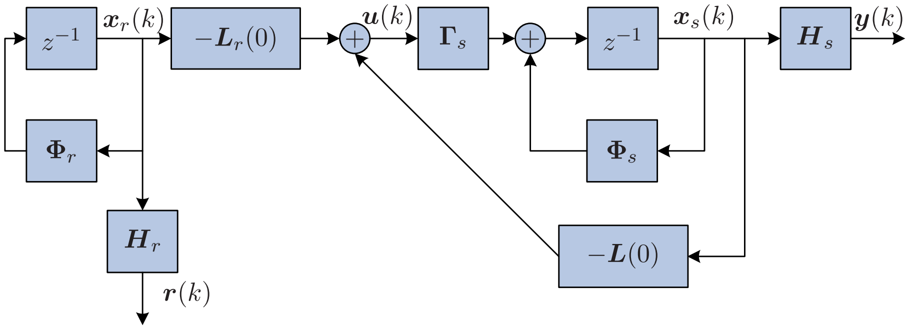


FIGURE 6.2: *The structure of the LQR controller with implemented reference states.*

6.2.3 Implementation of integral states

As stated in Section 4.1 it is chosen to implement integral states to remove steady state errors. The state-space representation of the integral is given as

$$\boldsymbol{x}_I(k+1) = \boldsymbol{x}_I(k) + \boldsymbol{e}(k) \tag{6.22}$$

$$\boldsymbol{e}(k) = \boldsymbol{r}(k) - \boldsymbol{y}(k) \ . \tag{6.23}$$

With the integral state the augmented state vector becomes

$$\boldsymbol{x}_{\text{aug}}(k) = \begin{bmatrix} \boldsymbol{x}_s^T(k) & \boldsymbol{x}_r^T(k) & \boldsymbol{x}_I^T(k) \end{bmatrix}^T \ , \tag{6.24}$$

which yields the new augmented system

$$\boldsymbol{x}_{\text{aug}}(k+1) = \left[\begin{array}{c|c|c} \boldsymbol{\Phi}_s & \boldsymbol{0} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{\Phi}_r & \boldsymbol{0} \\ \hline -\boldsymbol{H}_s & \boldsymbol{H}_r & \boldsymbol{I} \end{array}\right] \boldsymbol{x}_{\text{aug}}(k) + \begin{bmatrix} \boldsymbol{\Gamma}_s \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{u}(k) \tag{6.25}$$

$$= \boldsymbol{\Phi}_{\text{aug}} \boldsymbol{x}_{\text{aug}}(k) + \boldsymbol{\Gamma}_{\text{aug}} \boldsymbol{u}(k) \tag{6.26}$$

$$\boldsymbol{y}(k) = \begin{bmatrix} \boldsymbol{H}_s & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{x}_{\text{aug}}(k) = \boldsymbol{H}_y \boldsymbol{x}_{\text{aug}}(k) \tag{6.27}$$

$$\boldsymbol{e}(k) = \begin{bmatrix} -\boldsymbol{H}_s & \boldsymbol{H}_r & \boldsymbol{0} \end{bmatrix} \boldsymbol{x}_{\text{aug}}(k) = \boldsymbol{H}_e \boldsymbol{x}_{\text{aug}}(k) \tag{6.28}$$

$$\boldsymbol{x}_I(k) = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{x}_{\text{aug}}(k) = \boldsymbol{H}_I \boldsymbol{x}_{\text{aug}}(k) \ . \tag{6.29}$$

With the implemented integral state the performance index becomes

$$\mathcal{I} = \sum_{k=0}^{N-1} \left(\boldsymbol{x}_{\text{aug}}^T(k) \boldsymbol{Q}_1 \boldsymbol{x}_{\text{aug}}(k) + \boldsymbol{u}_{\text{aug}}^T(k) \boldsymbol{Q}_2 \boldsymbol{u}_{\text{aug}}(k) \right) + \boldsymbol{x}_{\text{aug}}^T(N) \boldsymbol{Q}_N \boldsymbol{x}_{\text{aug}}(N) \ , \tag{6.30}$$

where

$$\boldsymbol{Q}_1 = \boldsymbol{Q}_N = \boldsymbol{H}_e^T \boldsymbol{Q}_{1e} \boldsymbol{H}_e + \boldsymbol{H}_I^T(k) \boldsymbol{Q}_{1I} \boldsymbol{H}_I \tag{6.31}$$

Because of the implementation of integral action on all feedback states in the LQR controller, a total of 22 weight parameters for each of the hover and forward flight controller must be determined. With the augmented system given by $\boldsymbol{\Phi}_{\text{aug}}$ and $\boldsymbol{\Gamma}_{\text{aug}}$ and the design parameters $\boldsymbol{Q}_1$, $\boldsymbol{Q}_2$

and $\boldsymbol{Q}_N$, the optimal controller

$$\boldsymbol{u}(k) = -\boldsymbol{L}_{\text{aug}}(0)\boldsymbol{x}_{\text{aug}}(k) = -\begin{bmatrix} \boldsymbol{L} & \boldsymbol{L}_r & \boldsymbol{L}_I \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_s(k) \\ \boldsymbol{x}_r(k) \\ \boldsymbol{x}_I(k) \end{bmatrix} \quad , \tag{6.32}$$

with a structure as depicted in Figure 6.3 can be calculated by evaluating (6.7) and (6.8).

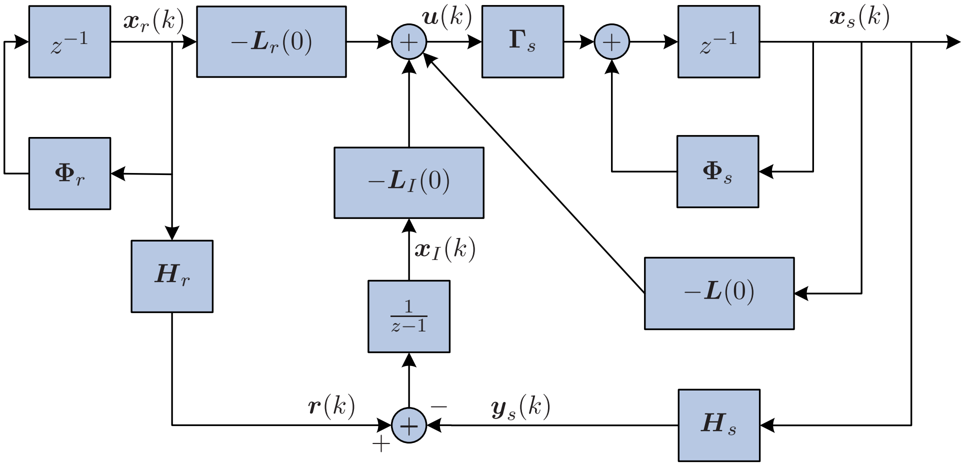


FIGURE 6.3: *The structure of the optimal controller with implemented reference and integral states.*

6.2.4 Modification of the LQR control structure

The current structure of the LQR controller is designed to weight the different references, which the controller must track. However, early controller designs showed that the gain matrices $\boldsymbol{L}_r$ and $\boldsymbol{L}$ are approximately identical with opposite sign ($\boldsymbol{L}_r = -\boldsymbol{L}$). By performing this substitution of $\boldsymbol{L}_r$ the following becomes clear:

$$\begin{aligned} \boldsymbol{u}(k) &= -\begin{bmatrix} \boldsymbol{L} & -\boldsymbol{L} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_s(k) \\ \boldsymbol{x}_r(k) \end{bmatrix} = \boldsymbol{L} \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_s(k) \\ \boldsymbol{x}_r(k) \end{bmatrix} \\ &= \boldsymbol{L}(\boldsymbol{x}_r(k) - \boldsymbol{x}_s(k)) = \boldsymbol{L}\boldsymbol{e}(k) \quad . \end{aligned} \tag{6.33}$$

From (6.33) it can be seen that the modification result in the controller trying to bring the error $\boldsymbol{e}(k)$ to zero. In addition, only two control gain matrices are needed to be designed.

Because the matrices $\boldsymbol{H}_r$ and $\boldsymbol{H}_s$ are identity matrices the controller structure depicted in Figure 6.4 is obtained. This structure will be used for implementation and test of the designed controllers in the nonlinear model in SIMULINK.

Having determined the design method and the actual structure for implementation of the LQR controllers the design of the hover and forward flight controllers can be performed, which is done in the following sections.

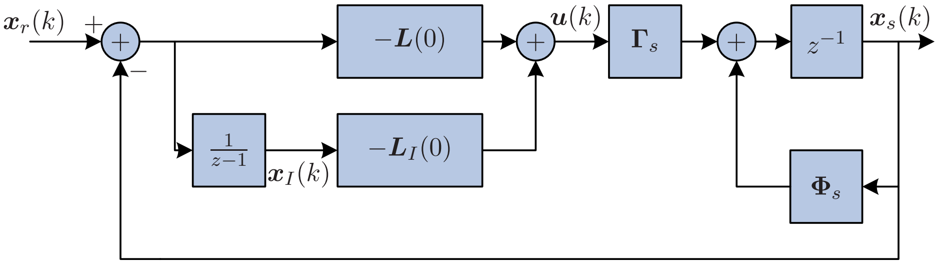


FIGURE 6.4: *The modified structure of the optimal controller.*

6.3 Hover controller design

In the following the design of the LQR hover controller will be performed. As described earlier the actual design lies in the determination of the weight matrices. Furthermore, it is described that if the open loop system is subject to limitations regarding the states and inputs, then the weight matrices can be initially be determined based on these limitations. This will be elaborated further in the following section.

6.3.1 Analytical weight determination

As described in Section 6.2.1 the weights can be determined (based on the limitations) as described by (6.9). The limitations in the system states can be determined based on open loop simulations of the nonlinear model. Because the model must operate in open loop it needs to be initialized in an equilibrium point regarding states and inputs. This equilibrium point is found by linearization of the nonlinear model in a desired operating point (see Appendix C for information on linearization). Note that it is not possible to operate the nonlinear model in open loop for all of these equilibria, as the linearization yields only approximated equilibria.

The limitations on the different states are determined as the highest values for which the nonlinear model does not destabilize. With this in mind the following state limitations have been identified by examining different flight velocities in forward and sideways direction:

$$\begin{aligned} {}^b\dot{x}_{\text{max}} &= 33\,\tfrac{\text{m}}{\text{s}} \quad \text{with pitch angle:} \quad {}^e\theta_{\text{min}} = -0.32\,\text{rad} \\ {}^b\dot{y}_{\text{max}} &= 20\,\tfrac{\text{m}}{\text{s}} \quad \text{with roll angle:} \quad {}^e\phi_{\text{max}} = 0.5\,\text{rad} \quad . \end{aligned} \tag{6.34}$$

The pitch angle ${}^e\theta_{min}$ is found as the lower limit due to its definition. The minimum vertical velocity ${}^b\dot{z}$ corresponding to upwards flight with ${}^e\theta = {}^e\phi = 0\,\text{rad}$ is by similar experiments identified as ${}^b\dot{z}_{\text{min}} = -12\,\frac{\text{m}}{\text{s}}$. The limitation on the yaw angle ${}^e\psi$ is intuitively not possible to determine, as there should not be a limit on the number of rotations possible to perform. Therefore, it is chosen as ${}^e\psi_{\text{max}} = 100\,\text{rad}$.

As with the translatory velocities, it is possible to linearize the nonlinear model to obtain an equilibrium point with a constant rotational velocity ${}^b\dot{\psi}$ about the z-axis in the BF. The limit has been identified as ${}^b\dot{\psi}_{max} = \pi\,\frac{\text{rad}}{\text{s}}$ corresponding to one revolution in two seconds. The angular velocities ${}^b\dot{\phi}$ and ${}^b\dot{\theta}$ are estimated to be equal to the limit on ${}^b\dot{\psi}$ yielding ${}^b\dot{\phi}_{\text{max}} = {}^b\dot{\theta}_{\text{max}} = \pi\,\frac{\text{rad}}{\text{s}}$.

Having determined the limits on the states the diagonal weight matrix in (6.35) is obtained for both the error and integral states. In addition, the weights on the control inputs are chosen as large values relative to the weights on the system states in order to obtain a conservative controller. The diagonal weight matrix for the control signals in (6.36) is chosen.

$$\begin{aligned}
\boldsymbol{Q}_{1e}(j,j) = \boldsymbol{Q}_{1I}(j,j) &= \left[\begin{array}{ccccccccc} \frac{1}{(0.5)^2} & \frac{1}{(-0.32)^2} & \frac{1}{100^2} & \frac{1}{33^2} & \frac{1}{20^2} & \frac{1}{-12^2} & \frac{1}{\pi^2} & \frac{1}{\pi^2} & \frac{1}{\pi^2} \end{array}\right] \\
&= \left[\begin{array}{ccccccccc} 4 & 9.7 & 0.0001 & 0.00091 & 0.0025 & 0.0069 & 0.10 & 0.10 & 0.10 \end{array}\right] \qquad (6.35) \\
\boldsymbol{Q}_2(j,j) &= \left[\begin{array}{cccc} 100 & 100 & 100 & 100 \end{array}\right] \qquad (6.36)
\end{aligned}$$

The time horizon parameter is chosen as $N = 3$, and the LQR hover controller is calculated as described in Section 6.2. The designed controller is tested by simulation on the nonlinear model initialized in the hover operating point and with the estimated states as feedback signals. The simulation result is depicted in Figure 6.5, where the hover operating point is illustrated as the red horizontal lines.

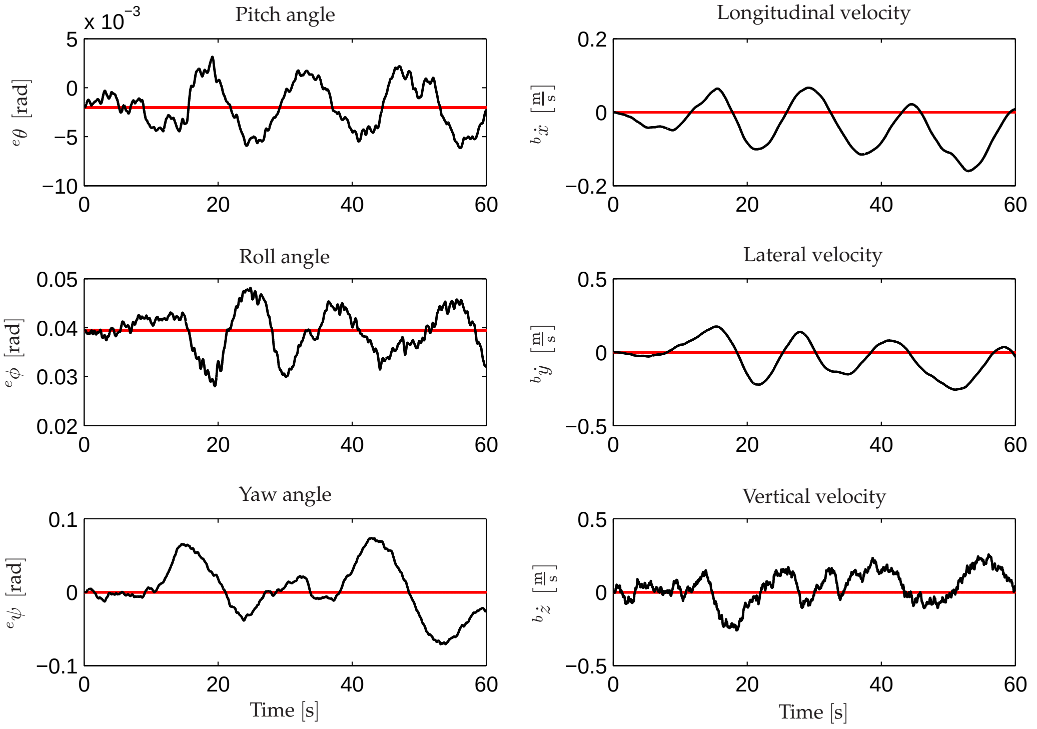


FIGURE 6.5: *Test result for the designed LQR hover controllers ability to stabilize the nonlinear model in hover. The red horizontal lines shows the hover operating point values.*

From the simulation of the hover controller it can be observed that it is capable of tracking the pitch angle set-point with a deviation of less than $5 \cdot 10^{-3}\,\mathrm{rad}$. A similar observation can be made for the roll angle, where a deviation of about $0.1\,\mathrm{rad}$ is determined. Taken the estimation signal and the analytical determined weight matrices into consideration, the performance of the hover controller is evaluated as relatively good because of it ability to track the set-point angles. However, the ability to track the pitch and roll angles can be interpreted as a result of reduced longitudinal and lateral velocity tracking ability. This can also be observed from Figure 6.5, where the longitudinal and lateral velocities have a stationary error as they approaches negative values over time. Furthermore, it can be observed that the designed controller is relatively slow taken the time scale into consideration. This means that compensation of disturbances applied to the UAV will happen slowly and may ultimately lead to instability. Therefore, a faster controller must be designed. In addition, as the ability to track velocity set-points in longitudinal and lateral directions are deemed more important than the tracking ability of the pitch and roll angles, it is necessary to improve the LQR controller by manually tuning the weight matrices.

6.3.2 Weight tuning regarding stationary properties

The tuning of the weight matrices is performed iteratively using trial-and-error during multiple simulations to obtain the weights seen in (6.37) –(6.39):

$$\boldsymbol{Q}_{1e}(j,j) = [\ 0.016 \quad 0.81 \quad 1.0 \quad 1.5 \quad 0.0044 \quad 1.0 \quad 0.036 \quad 0.18 \quad 0.00091\ ] \tag{6.37}$$

$$\boldsymbol{Q}_{1I}(j,j) = [\ 0.048 \quad 1.2 \quad 3.6 \quad 0.31 \quad 0.013 \quad 1.5 \quad 0.0010 \quad 0.10 \quad 0.0027\ ] \tag{6.38}$$

$$\boldsymbol{Q}_{2}(j,j) = [\ 3.2 \cdot 10^{3} \quad 17 \quad 1.4 \cdot 10^{2} \quad 2.7\ ]\ . \tag{6.39}$$

In addition, the time horizon is changed to $N = 5$ in order to improve the stationary properties in the closed loop. This is the final value of N and will therefore not be discussed any further throughout this thesis.

A test simulation using the new weight matrices is performed with the result depicted in Figure 6.6. Compared to the results from Figure 6.5 the roll and pitch angles obtain larger variations, which is a trade-off that leads to better tracking of the longitudinal and lateral velocities. This is explained by the fact, that when the UAV needs to gain e.g. longitudinal velocity, this is done by altering the current pitch angle. It is observed that the angles ${}^{e}\theta$, ${}^{e}\phi$ and ${}^{e}\psi$ varies less than $0.1\,\mathrm{rad}$ from their respective set-points, and the three translatory velocities ${}^{b}\dot{x}$, ${}^{b}\dot{y}$ and ${}^{b}\dot{z}$ varies less than $0.3\,\frac{\mathrm{m}}{\mathrm{s}}$. The size of these stationary deviations are estimated as being small enough for the controller to fulfill their purpose of keeping the UAV in the condition hover. Furthermore, it can be observed that the hover controller has become become a faster controller than the previous, therefore it is decided to use this for control of the system.

Having validated the designed controller with respect to its stationary properties, the dynamical properties of the controller must be tested. Furthermore, if deemed necessary the controller will be tuned based on the observations done by dynamical tests of the closed loop system.

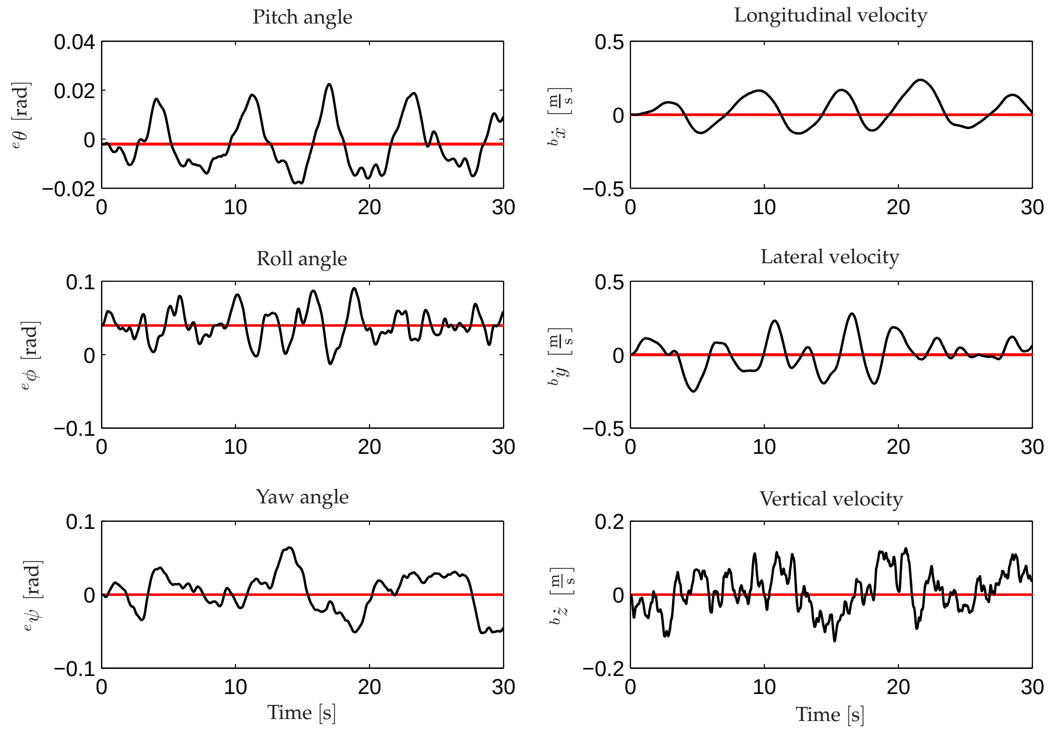


FIGURE 6.6: *Test result of the improved LQR controller. As it can be observed the variations of the angles are lager than the previous designed controller. However, the new controller has better overall tracking ability and perform faster control.*

6.3.3 Weight tuning regarding dynamical properties

The dynamical properties of the designed controller are examined by performing offset tests and analyzing the controllers ability to return the UAV to the hover operating point. The test is performed by initializing the nonlinear model in hover with the offsets listed in Table 4.1 on page 22 applied to the initial states. For this purpose the UAV is considered as being in hover when the angles varies less than $0.1\,\mathrm{rad}$ and the velocities varies less than $0.3\,\frac{\mathrm{m}}{\mathrm{s}}$ from the hover operating point (values obtained from Figure 6.6). As it is not possible to determine some maximum time for the controller to stabilize the UAV, it is instead required to settle the six control states equally fast, with a maximum time dispersion of 5 seconds.

The simulation result is depicted in Figure 6.7. It is observed that the time for settling the longitudinal velocity at the stationary limits (indicated by dashed lines) is approximately 23 seconds, where it only takes approximately 6 seconds to settle the pitch angle. An intuitive way for obtaining a lower settling time for the longitudinal velocity is to increase the corresponding weights for ${}^{b}\dot{x}$ in $\boldsymbol{Q}_{1e}$ and/or $\boldsymbol{Q}_{1I}$. A few trial-and-error simulations with increased weights for ${}^{b}\dot{x}$ quickly

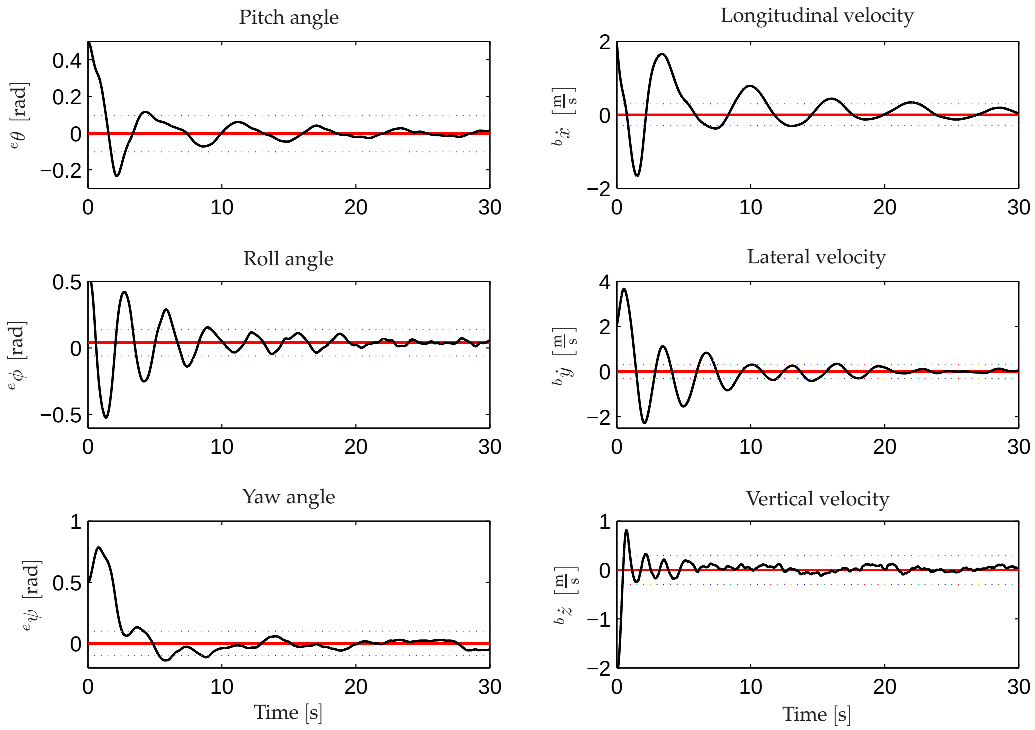


FIGURE 6.7: *Dynamical test of the improved LQR hover controller. The large variations in the roll angle causes significant variations in the lateral velocity as well.*

reveal, that this approach is not the solution. The explanation is found in the severe cross coupling between the pitch angle $^e\theta$ and the longitudinal velocity $^b\dot{x}$. From observation of Figure 6.7 it becomes clear that although $^e\theta$ settles within 6 seconds, it does vary throughout the following 20 seconds, which causes $^b\dot{x}$ to settle relative late. Therefore, the focus must be turned to tuning the weight for $^e\theta$. Similar oscillating behavior is observed for the roll angle and the lateral and vertical velocities.

Consequently some of the weight elements of the weight matrices in (6.37) and (6.38) have been multiplied by the following factors:

$$\begin{aligned} &\boldsymbol{Q}_{1e}(1,1) = 10 \quad \boldsymbol{Q}_{1e}(2,2) = 10 \quad \boldsymbol{Q}_{1e}(3,3) = 20 \\ &\boldsymbol{Q}_{1I}(1,1) = 2 \quad\;\; \boldsymbol{Q}_{1I}(2,2) = 2\;\;, \end{aligned} \tag{6.40}$$

which result in the dynamical properties depicted in Figure 6.8 for the LQR hover controller. It is observed that the six control states settle in the time period $2\,\mathrm{s} < t < 6\,\mathrm{s}$, which satisfies the requirement of a maximum dispersion of 5 seconds. In addition, it is observed that the stationary properties of the LQR controller have also improved compared to Figure 6.7.

As the LQR hover controller is capable of stabilizing the UAV in the hover operating point

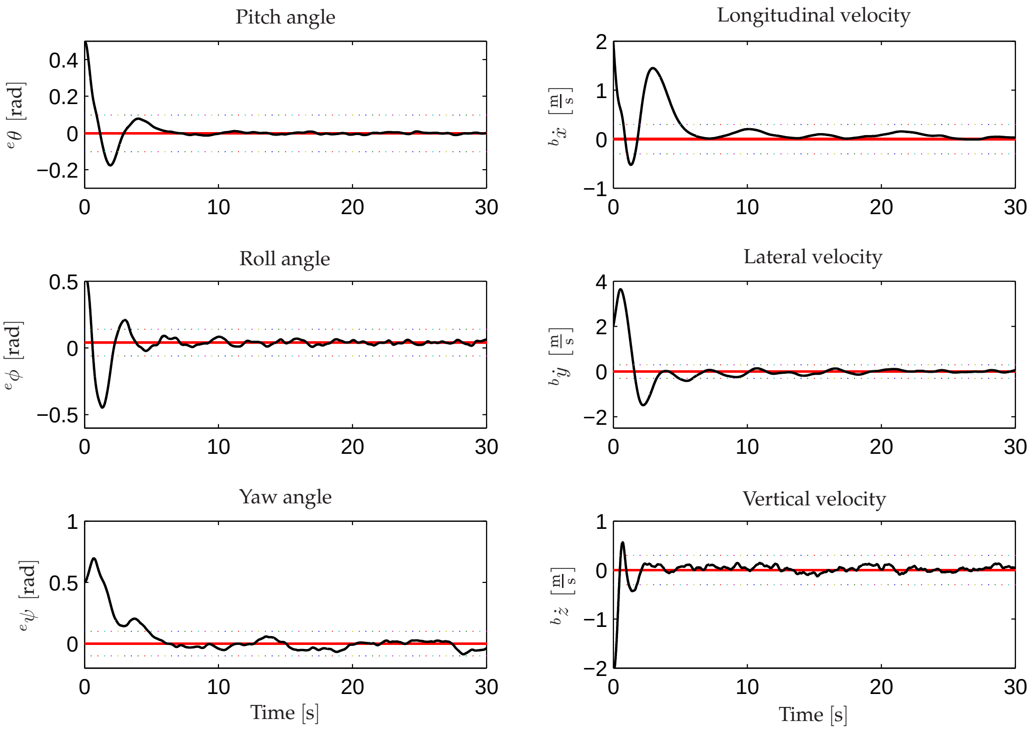


FIGURE 6.8: *Dynamical test for the final LQR hover controller. It is seen that the control states settles faster and with less overshoot with respect to the former controller weights.*

(given the initial offset values) within 6 seconds the stationary and dynamical properties of the controller are concluded as satisfactory for further use in the gain scheduling controller.

In the next section the classic SISO hover and the optimal LQR hover controller are compared regarding stationary and dynamical properties.

6.3.4 Comparison of classic and LQR hover control

As described in the introduction to this thesis the classic control strategy is applied to be able to compare it with more advanced control strategies, here optimal LQR control. The comparison of the two control strategies is based on the individual tests performed with respect to stationary and dynamical properties for each of the two controllers.

The first test performed for each controller concerns the stationary properties. For the LQR controller a new simulation different from the one in Section 6.3.2 is performed, as the weight matrices for this controller were changed during evaluation of the dynamical properties. This new simulation result is depicted as the blue graphs in Figure 6.9 together with the test result obtained for the classic controller in Section 5.4 (the green graphs).

The graphs for the yaw angles looks much alike which indicates, that the two controllers performs equally with respect to the yaw angle, and that the variations to a great extent are a result of the estimated yaw angle being different from the actual yaw angle for the model.

It is observed that the pitch angle and the longitudinal and vertical velocities are more resonant for the classic controller, which is an unwanted property for controlling a helicopter. The LQR controller is better than the classic controller for controlling these, as it maintains the states closer to the references and less resonant. The classic controller seems to control the lateral velocity better than the LQR controller. However, as the main part of the variations of the graphs, as with the yaw angle, results from the estimated state being different from the real state, no conclusion can be drown regarding which controller is the better for controlling the lateral velocity. It is concluded that the LQR controller overall has better stationary properties than the classic controller.

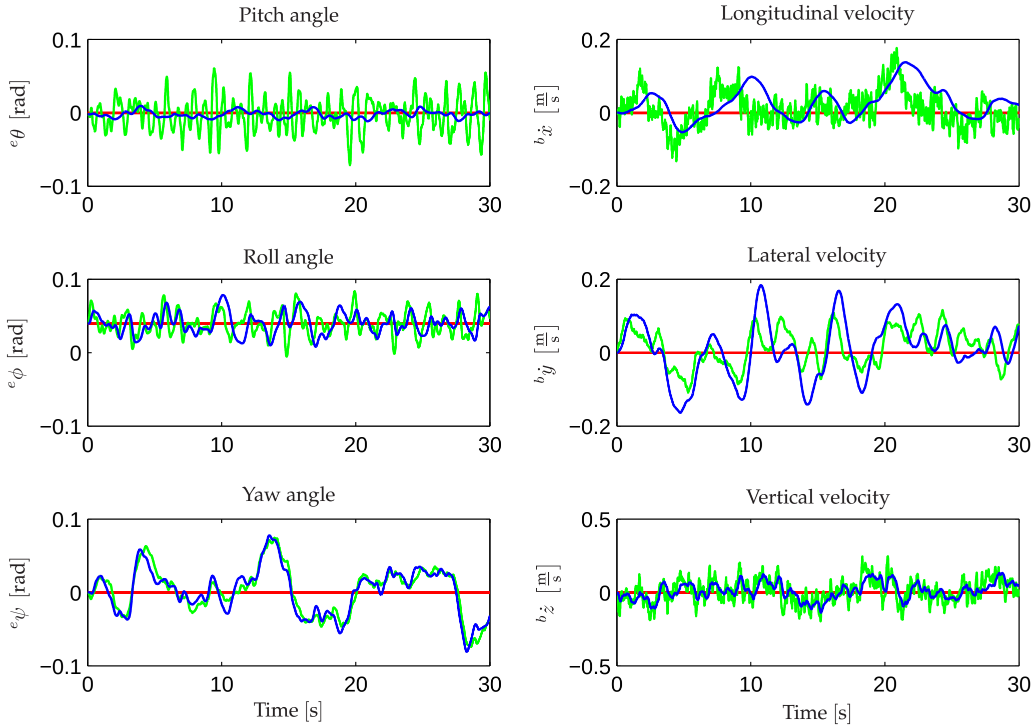


FIGURE 6.9: *Comparison of the classic (green) and LQR (blue) controllers for hover with respect to the stationary properties. It is observed that the pitch angle and the longitudinal and vertical velocities are more resonant for the classic controller.*

For both controllers tests regarding the dynamical properties has been performed as well. The nonlinear model was initialized with offsets in the states $^e\phi, {}^e\theta, {}^e\psi, {}^b\dot{x}, {}^b\dot{y}$ and $^b\dot{z}$, and the ability of the controllers to settle these six states to the levels shown in Figure 6.9 was examined in terms of time spend. The classic controller used $10\,\mathrm{s}$ and the LQR controller $5-6\,\mathrm{s}$. It is therefore concluded, that the LQR controller has better dynamical properties than the classic controller.

In the following section the design of the LQR forward flight controller is described.

6.4 Forward flight controller design

As a starting point for the design of the diagonal weight matrices for the forward flight controller, the final weights determined for the hover controller are used. The forward flight controller is afterwards tuned with respect to the stationary properties using trial-and-error to obtain the following weight matrices:

$$\boldsymbol{Q}_{1e}(j,j) = [\ 1.3 \quad 8.1 \quad 20 \quad 0.15 \quad 0.079 \quad 0.10 \quad 0.036 \quad 0.019 \quad 0.0001\ ] \tag{6.41}$$

$$\boldsymbol{Q}_{1I}(j,j) = [\ 0.024 \quad 1.8 \quad 3.6 \quad 0.022 \quad 0.0080 \quad 0.15 \quad 0.0011 \quad 0.011 \quad 0.00027\ ] \tag{6.42}$$

$$\boldsymbol{Q}_{2}(j,j) = [\ 295 \quad 2.1 \quad 21 \quad 0.33\ ]\ . \tag{6.43}$$

For tuning both the hover and the forward flight LQR controller some identified rules of thumb have been exploited in addition to the trial-and-error method. As described in the previous section the strong cross coupling between ${}^{b}\dot{x}$ and ${}^{e}\theta$ resulted in changing weights for ${}^{e}\theta$ instead of ${}^{b}\dot{x}$ for the hover controller. In general, the knowledge on cross coupling in the nonlinear model described by the input/output relations divided into four groups in Section 2.3 have been used for weight tuning.

6.4.1 Stationary properties

A stationary test simulation of the nonlinear model initialized in the $10\,\frac{\mathrm{m}}{\mathrm{s}}$ forward flight operating point and stabilized by the forward flight controller is depicted in Figure 6.10. It is observed that the lateral velocity deviates from the reference of $0\,\frac{\mathrm{m}}{\mathrm{s}}$ with a peak of $1.5\,\frac{\mathrm{m}}{\mathrm{s}}$, which is significant compared to the stationary deviations of less than $0.3\,\frac{\mathrm{m}}{\mathrm{s}}$ observed for the hover controller (see Section 6.3.2). The explanation is, that some of the estimated states used as feedback for control purposes have a significant deviation from the real state. In this case the deviation on the lateral velocity origins from poor estimates of the yaw angle ${}^{e}\psi_{\text{est}}$ and lateral velocity ${}^{b}\dot{y}_{\text{est}}$. This is shown in Figure 6.11, where the test simulation is repeated using the real states as feedback. It is observed from Figure 6.11 that the controller is able to stabilize both ${}^{e}\psi$ and ${}^{b}\dot{y}$ at approximately zero in contrast to their respective estimates. Furthermore, it is observed that the yaw angle estimate has a positive error of up to $0.25\,\mathrm{rad}$ and the lateral velocity a negative error of up to $2.5\,\frac{\mathrm{m}}{\mathrm{s}}$. Another important observation is the initial increase and decrease in the longitudinal velocity and pitch angle respectively. This is due to the initial value of the estimated longitudinal velocity, which at all times initially is zero. This leads to an initial error input to the controller of $10\,\frac{\mathrm{m}}{\mathrm{s}}$ even though the real error (with respect to the real initial state) is zero.

As this thesis does not concern sensor equipments and sensor fusion and estimation, the significant estimation error in ${}^{e}\psi_{\text{est}}$ and ${}^{b}\dot{y}_{\text{est}}$ will not be treated any further in this thesis. As simulations

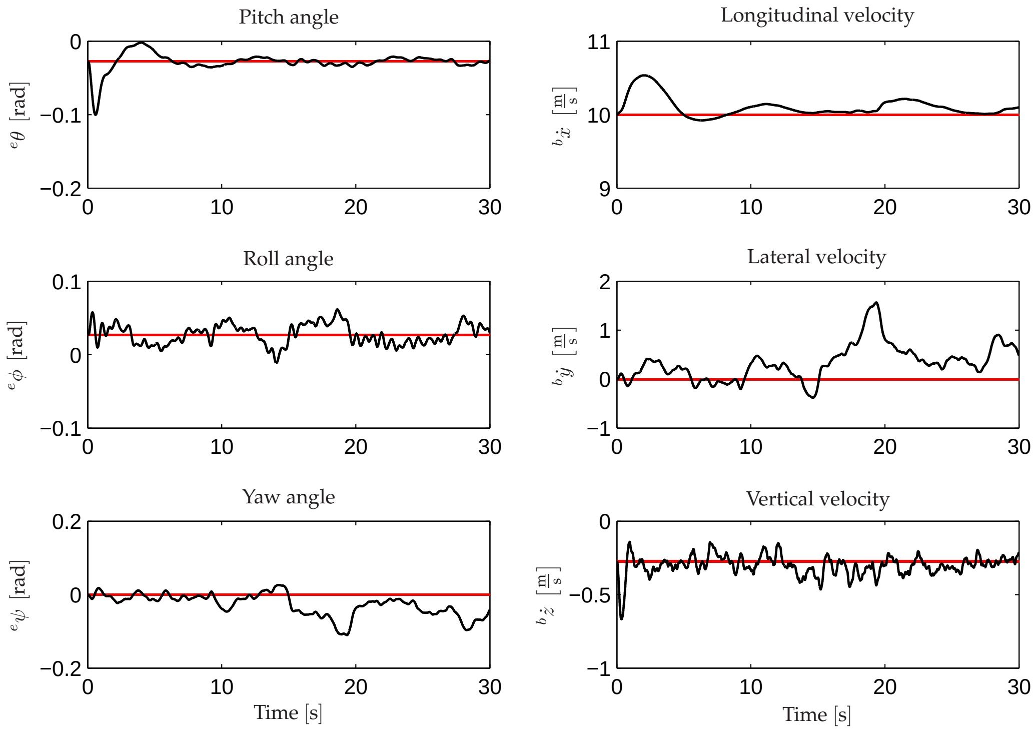


FIGURE 6.10: *Stationary simulation results of the LQR forward flight controller initialized in the forward flight operating point of* $10\ \frac{\mathrm{m}}{\mathrm{s}}$.

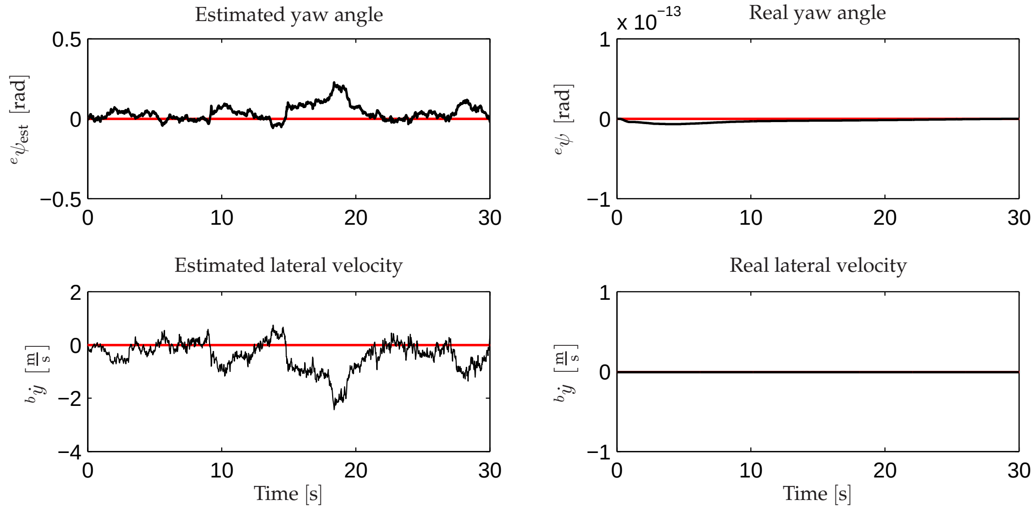


FIGURE 6.11: *Simulation of the forward flight controller in the operating point of* $10\ \frac{\mathrm{m}}{\mathrm{s}}$ *using the real states as feedback signals.*

over a greater period of time show, that the estimation errors does not become larger, and as the top level supervisory controller provides position control in the control system. Furthermore, the drift effect from the observed stationary lateral velocity error is assumed not to have noticeable impact on the performance of the control system. Based on this assumption the designed LQR forward flight controller it is assessed that it has satisfactory stationary performance in the forward flight operating point.

In the following section the dynamical properties of the designed forward flight controller will be evaluated.

6.4.2 Dynamical properties

The dynamical properties of the LQR forward flight controller are examined in the same manner as for the LQR hover controller, but with starting point in the forward flight operating point. The result of the test is depicted in Figure 6.12, where it is observed that four of the controlled outputs settle to their respective references; ${}^e\theta$, ${}^b\dot{x}$, ${}^e\phi$ and ${}^b\dot{z}$. Regarding the remaining two control states (${}^b\dot{y}$ and ${}^e\psi$) the same behavior as described in the previous section can be observed. However,

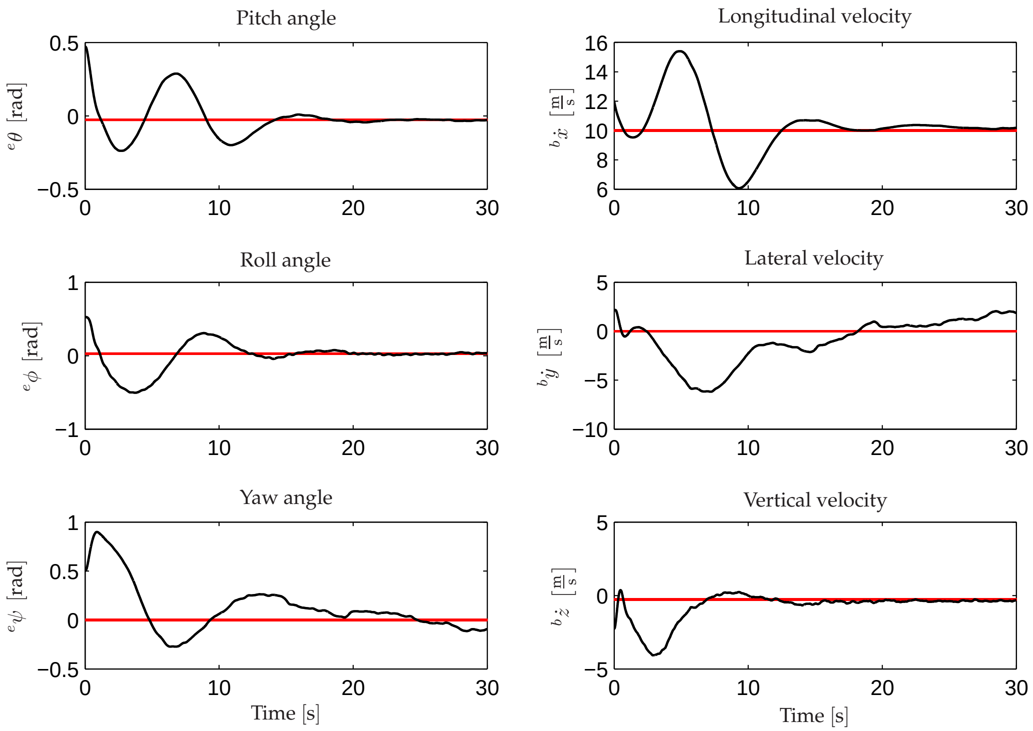


FIGURE 6.12: *Validation test for the LQR forward flight controller where an initial offset is given on six of the controlled outputs; ${}^e\theta$, ${}^b\dot{x}$, ${}^e\phi$, ${}^b\dot{y}$, ${}^e\psi$ and ${}^b\dot{z}$.*

since it has been concluded that the deviations in ${}^{b}\dot{y}$ and ${}^{e}\psi$ are acceptable this is overlooked for the dynamical test as well.

However, another important observation is that of the rather large variations during the first 15 seconds of the different control states. This is mainly due to the initial values of the estimated states, which are all zero. Consequently the controller compensates on a wrong error of $10\,\frac{\mathrm{m}}{\mathrm{s}}$ in the longitudinal velocity, where the actual error on the model state $-2\,\frac{\mathrm{m}}{\mathrm{s}}$. The integral action thereby accumulates a large error, which helps explaining the 15 seconds of large variation.

With the above in mind the designed LQR forward flight controller is assessed to have satisfactory dynamical performance because it is capable of returning the UAV to its forward flight operating point. Therefore, this controller will be used in the gain scheduling controller, which is treated in the next chapter.

7 Observer-based gain scheduling control

This chapter explains the design and implementation of gain scheduling into the control system. The implementation of gain scheduling makes it possible to switch between several controllers designed for one or more operating point(s). In addition, the use of gain scheduling enables the helicopter to achieve a more reliable and versatile manoeuvrability when performing level 1 of the IARC.

7.1 Gain scheduling method

As mentioned earlier in this thesis the implemented controllers are designed based on linearizations of the nonlinear model in different operating points. Here the operating points have been chosen as hover and forward flight of $10\,\frac{\mathrm{m}}{\mathrm{s}}$, and linear controllers have been developed for the respective operating points. The challenge is to switch between these two specific controllers without compromising overall system stability.

Three ways to approach the problem of switching between controllers are; direct switch between controllers, gradual transition using a controller weight function α_{gs}, and an observer-based gain scheduling version of the weighted controller method with improved stability properties.

7.1.1 Direct controller switch

Direct switch between controllers is the most simple method of gain scheduling. This method is executed simply by replacing one controller with another. Consider the optimal LQR controller form

$$\boldsymbol{u}(t) = \boldsymbol{L}_0\boldsymbol{e}(t) - \boldsymbol{L}_{I0}\boldsymbol{x}_I(t) \ \ , \tag{7.1}$$

where $\boldsymbol{u}(t)$ is the system control signal, $\boldsymbol{e}(t)$ is the control error and $\boldsymbol{x}_I(t)$ is the integral state with $\dot{\boldsymbol{x}}_I(t) = \boldsymbol{x}_I(t) + \boldsymbol{e}(t)$. $\boldsymbol{L}_0$ and $\boldsymbol{L}_{I0}$ are the optimal proportional and integral gains respectively for a given operating point. Now assume that a switch is performed from the first set of controllers ($\boldsymbol{L}_0$ and $\boldsymbol{L}_{I0}$) to a second set of controllers ($\boldsymbol{L}_1$ and $\boldsymbol{L}_{I1}$) at a given time instance $\bar{t}$. If $\boldsymbol{L}_1$ is different from $\boldsymbol{L}_0$ and/or $\boldsymbol{L}_{I1}$ is different from $\boldsymbol{L}_{I0}$, which is probably the case if the controller sets are designed for different operating points, the value of $\boldsymbol{u}(t)$ will be subject to a momentary change with a size dependent on the difference between the controller sets, unless both $\boldsymbol{e}(t)$ and $\boldsymbol{x}_I(t)$ at the time $\bar{t}$ are both zero. Hence, this method may cause the control system to become unstable.

7.1.2 Weighted controller switch

A less bumpy transfer between two controller sets are achieved by implementing a weight function α_{gs}. The general control law for this method is

$$\boldsymbol{u}(t) = (\alpha_{gs}\boldsymbol{L}_1 + (1 - \alpha_{gs}\boldsymbol{L}_0))\,\boldsymbol{e}(t) + (\alpha_{gs}\boldsymbol{L}_{I1} + (1 - \alpha_{gs})\boldsymbol{L}_{I0})\,\boldsymbol{x}_I(t) \quad , \tag{7.2}$$

where $\alpha_{gs} \in [0; 1]$. By changing α_{gs} gradually from $0 \rightarrow 1$, (7.2) will result in a switch from the first controller set ($\boldsymbol{L}_0$ and $\boldsymbol{L}_{I0}$) to the second set ($\boldsymbol{L}_1$ and $\boldsymbol{L}_{I1}$), where a smooth control signal $\boldsymbol{u}(t)$ (no momentary changes) is maintained, as long as $\boldsymbol{e}(t)$ and $\boldsymbol{x}_I(t)$ does not change momentary either. However, it is important to note that this form of gain scheduling between to sets of controllers does not guarantee stability of the control system for $0 < \alpha_{\text{gs}} < 1$, Bendtsen et al. [2005].

7.1.3 Controller switch using observer-based gain scheduling

In Bendtsen et al. [2005] a controller construction that guarantees stability for any $\alpha_{\text{gs}} \in [0; 1]$ when applied to one linear system is introduced. It is claimed, that while it does not guarantee stability between two operating points, it is still an improvement compared to (7.2). This controller construction is implemented and tested on the nonlinear model of the UAV in this thesis.

Starting point is taken in an observer based structure mapping an input signal $\boldsymbol{u}(t) \in \mathbb{R}^m$ to an output signal $\boldsymbol{y}(t) \in \mathbb{R}^p$, which can be described as

$$\boldsymbol{H} = \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{C} & \boldsymbol{D} \end{array}\right] \quad , \tag{7.3}$$

which represent a standard state space model

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) \end{aligned} \quad . \tag{7.4}$$

The above system can be extended to a so called *two port system*

$$\mathcal{H} = \left[\begin{array}{c|cc} \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{B}_2 \\ \hline \boldsymbol{C} & \boldsymbol{D} & \boldsymbol{D}_{12} \\ \boldsymbol{C}_2 & \boldsymbol{D}_{21} & \boldsymbol{D}_{22} \end{array}\right] \quad , \tag{7.5}$$

which, if interconnected with a *zero system*, yields

$$\left[\begin{array}{cc} \boldsymbol{I} & \boldsymbol{0} \end{array}\right] \mathcal{H} \left[\begin{array}{c} \boldsymbol{I} \\ \boldsymbol{0} \end{array}\right] = \mathcal{H} \star \boldsymbol{0} = \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{C} & \boldsymbol{D} \end{array}\right] = \boldsymbol{H} \quad . \tag{7.6}$$

mapping two input vectors to two output vectors. The $\star$ in (7.6) represent the *Redheffer Star-product*, where the calculation method for the interconnection shown in Figure 7.1 between two two port systems

$$\mathcal{P} = \left[\begin{array}{c|cc} \boldsymbol{A} & \boldsymbol{B}_1 & \boldsymbol{B}_2 \\ \hline \boldsymbol{C}_1 & \boldsymbol{D}_{11} & \boldsymbol{D}_{12} \\ \boldsymbol{C}_2 & \boldsymbol{D}_{21} & \boldsymbol{D}_{22} \end{array}\right] , \quad \mathcal{K} = \left[\begin{array}{c|cc} \boldsymbol{A}_K & \boldsymbol{B}_{K1} & \boldsymbol{B}_{K2} \\ \hline \boldsymbol{C}_{K1} & \boldsymbol{D}_{K11} & \boldsymbol{D}_{K12} \\ \boldsymbol{C}_{K2} & \boldsymbol{D}_{K21} & \boldsymbol{D}_{K22} \end{array}\right]$$

is defined as (see Zhou et al. [1996]):

$$\mathcal{S}(\mathcal{P},\mathcal{K}) = \left[\begin{array}{c|c} \bar{A} & \bar{B} \\ \hline \bar{C} & \bar{D} \end{array}\right] \quad ,$$

where

$$\bar{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A} + \boldsymbol{B}_2\tilde{\boldsymbol{R}}^{-1}\boldsymbol{D}_{K11}\boldsymbol{C}_2 & \boldsymbol{B}_2\tilde{\boldsymbol{R}}^{-1}\boldsymbol{C}_{K1} \\ \boldsymbol{B}_{K1}\boldsymbol{R}^{-1}\boldsymbol{C}_2 & \boldsymbol{A}_K + \boldsymbol{B}_{K1}\boldsymbol{R}^{-1}\boldsymbol{D}_{22}\boldsymbol{C}_{K1} \end{bmatrix} \tag{7.7}$$

$$\bar{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{B}_1 + \boldsymbol{B}_2\tilde{\boldsymbol{R}}^{-1}\boldsymbol{D}_{K11}\boldsymbol{D}_{21} & \boldsymbol{B}_2\tilde{\boldsymbol{R}}^{-1}\boldsymbol{D}_{K12} \\ \boldsymbol{B}_{K1}\boldsymbol{R}^{-1}\boldsymbol{D}_{21} & \boldsymbol{B}_{K2} + \boldsymbol{B}_{K1}\boldsymbol{R}^{-1}\boldsymbol{D}_{22}\boldsymbol{D}_{K12} \end{bmatrix} \tag{7.8}$$

$$\bar{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{C}_1 + \boldsymbol{D}_{12}\boldsymbol{D}_{K11}\boldsymbol{R}^{-1}\boldsymbol{C}_2 & \boldsymbol{D}_{12}\tilde{\boldsymbol{R}}^{-1}\boldsymbol{C}_{K1} \\ \boldsymbol{D}_{K21}\boldsymbol{R}^{-1}\boldsymbol{C}_2 & \boldsymbol{C}_{K2} + \boldsymbol{D}_{K21}\boldsymbol{R}^{-1}\boldsymbol{D}_{22}\boldsymbol{C}_{K1} \end{bmatrix} \tag{7.9}$$

$$\bar{\boldsymbol{D}} = \begin{bmatrix} \boldsymbol{D}_{11} + \boldsymbol{D}_{12}\boldsymbol{D}_{K11}\boldsymbol{R}^{-1}\boldsymbol{D}_{21} & \boldsymbol{D}_{21}\tilde{\boldsymbol{R}}^{-1}\boldsymbol{D}_{K12} \\ \boldsymbol{D}_{K21}\boldsymbol{R}^{-1}\boldsymbol{D}_{21} & \boldsymbol{D}_{K22} + \boldsymbol{D}_{K21}\boldsymbol{R}^{-1}\boldsymbol{D}_{22}\boldsymbol{D}_{K12} \end{bmatrix} \tag{7.10}$$

$$\boldsymbol{R} = \boldsymbol{I} - \boldsymbol{D}_{22}\boldsymbol{D}_{K11}, \quad \tilde{\boldsymbol{R}} = \boldsymbol{I} - \boldsymbol{D}_{K11}\boldsymbol{D}_{22} \quad .$$

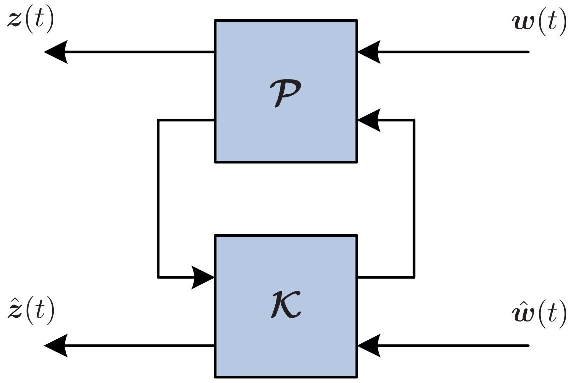


FIGURE 7.1: *Illustration of the ⋆ interconnection between two two port systems.*

The above described method for interconnection between systems is combined with the weighted controller switch method to perform observer-based gain scheduling between the two LQR controllers developed in Chapter 6. In the following section the matrices associated with the implementation of the observer-based gain scheduling are explained.

7.2 Observer-based gain scheduling design

The design presented throughout this section is based on the work described in Bendtsen et al. [2005]. It is presented in continuous time, however it can be extended into discrete time by simple discretizations of the involved matrices. The time index (t) is omitted to ease the notation.

For the purpose of observer-based gain scheduling it is necessary to design observers for the two operating points; hover, and forward flight of $10\ \frac{\mathrm{m}}{\mathrm{s}}$. The design of the observers follow the general observer design theory, where any given system is simplified by linearization, and an observer feedback gain $\boldsymbol{F}$ is determined such that the output of the observer $\hat{\boldsymbol{y}}$ converges to the sensor output $\boldsymbol{y}$ of the system as illustrated in Figure 7.2.

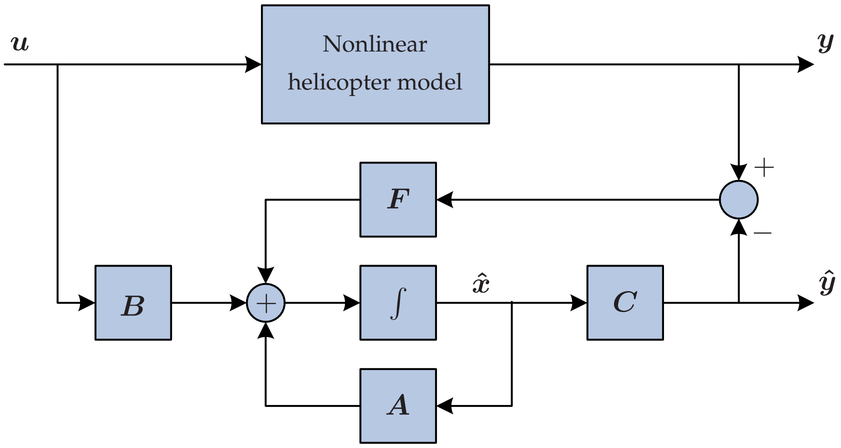


FIGURE 7.2: *Illustration of the general observer theory with observer feedback gain.*

The system matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are determined as described in Appendix C for the hover ($\boldsymbol{A}_0$ and $\boldsymbol{B}_0$) and the forward flight ($\boldsymbol{A}_1$ and $\boldsymbol{B}_1$) operating points. The output matrices $\boldsymbol{C}_0$ and $\boldsymbol{C}_1$ are both identity ($\boldsymbol{I}$) with the same dimension as $\boldsymbol{A_0}$ and $\boldsymbol{A_1}$, as all states can be measured directly and a full state observer is chosen. The observer feedback gains are therefore the only remaining parts for implementing the observer structure in Figure 7.2. Note that there must be designed a specific observer for each of the two operating points, which will be elaborated on later. For now the observer gains are denoted with $\boldsymbol{F}_0$ and $\boldsymbol{F}_1$ for the hover and the forward flight operating points respectively.

The goal for the observer design is to construct a two port system

$$\begin{bmatrix}\dot{\hat{\boldsymbol{x}}}_0\\ \dot{\hat{\boldsymbol{x}}}_{I0}\\ \boldsymbol{u}\\ \boldsymbol{e}_q\end{bmatrix} = \overbrace{\left[\begin{array}{cc|cc} \boldsymbol{A}_0-\boldsymbol{B}_0\boldsymbol{L}_0-\boldsymbol{F}_0\boldsymbol{C}_0 & \boldsymbol{B}_0 & \boldsymbol{F}_0 & \boldsymbol{B}_0\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L}_{I0} & r\boldsymbol{I}\\ \hline -\boldsymbol{L}_0 & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I}\\ -\boldsymbol{C}_0 & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0}\end{array}\right]}^{\boldsymbol{\mathcal{K}}} \begin{bmatrix}\hat{\boldsymbol{x}}_0\\ \hat{\boldsymbol{x}}_{I0}\\ \boldsymbol{e}\\ \boldsymbol{u}_q\end{bmatrix}, \tag{7.11}$$

with the structure depicted in Figure 7.3, where $r > 0$ represents the integrator factorization and $\boldsymbol{L}_0$ and $\boldsymbol{L}_{I0}$ represents the LQR hover controller. The LQR controller matrices of dimension $\mathbb{R}^{4x9}$ are each extended with three columns of zeros as feedback gains for the position errors of ${}^e x$, ${}^e y$ and ${}^e z$ to fit the full state observer design. The augmented observer state vector is defined as $\hat{\boldsymbol{\xi}}_0 = [\ \hat{\boldsymbol{x}}_0^T \quad \hat{\boldsymbol{x}}_{I0}^T\ ]^T$. As it can be seen from (7.11) there are two input and two output vectors, where $\boldsymbol{\mathcal{K}}$ takes the input vectors $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{y}_{\text{ref}}$ and $\boldsymbol{u}_q$, and yields the output vectors $\boldsymbol{u}$ and

$\boldsymbol{e}_q = \boldsymbol{e} - \boldsymbol{C}_0\hat{\boldsymbol{x}}_0$. Notice that the observers about to be designed are not observing the actual states but the state error.

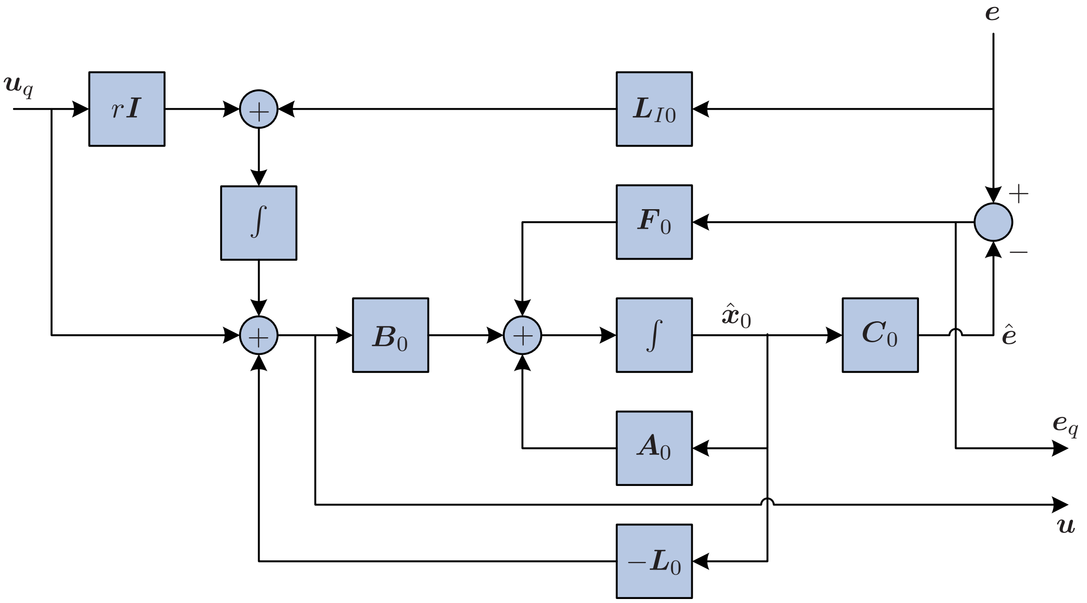


FIGURE 7.3: *Graphical illustration of the observer-based controller given by* (7.11) *(illustrated in continues time).*

By performing interconnection between $\mathcal{K}$ and a zero system it can be shown that the resulting system becomes a standard observer-based controller $\boldsymbol{K}_0$ with integral action

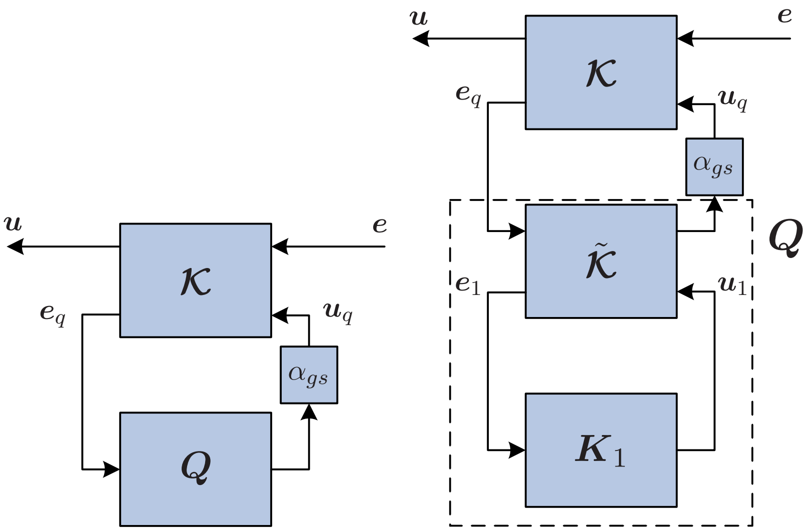
$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \mathcal{K} \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix} = \mathcal{K} \star \boldsymbol{0} = \left[\begin{array}{cc|c} \boldsymbol{A}_0 - \boldsymbol{B}_0\boldsymbol{L}_0 - \boldsymbol{F}_0\boldsymbol{C}_0 & \boldsymbol{B}_0 & \boldsymbol{F}_0 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L}_{I0} \\ \hline -\boldsymbol{L}_0 & \boldsymbol{I} & \boldsymbol{0} \end{array}\right] = \boldsymbol{K}_0 \ , \tag{7.12}$$

which is equal to $\alpha_{\text{gs}} = 0$ in

$$\boldsymbol{K}(\alpha_{\text{gs}}\boldsymbol{Q}) = \mathcal{K} \star (\alpha_{\text{gs}}\boldsymbol{Q}) \tag{7.13}$$

where $\boldsymbol{K}$ is the total interconnected system in Figure 7.4(a). Having determined the structure for the two port system for the first observer-based controller $\mathcal{K}$ the next step is to identify the structure for the scheduler $\boldsymbol{Q}$. Because $\boldsymbol{Q}$ is the control system scheduler it must by definition contain the second controller to switch to. Furthermore, it is desired to decouple the first controller completely when $\alpha_{gs} = 1$ and therefore it is necessary to design a two port system $\tilde{\mathcal{K}}$ with the state vector $\tilde{\boldsymbol{\xi}} = [\ \tilde{\boldsymbol{x}}^T \quad \tilde{\boldsymbol{x}}_I^T\ ]^T$ that, when $\alpha_{\text{gs}} = 1$, yields an identity system when interconnected with $\mathcal{K}$. By having an identity system of $\mathcal{K} \star \tilde{\mathcal{K}}$ it is possible to implement any given second observer-based controller $\boldsymbol{K}_1$ interconnected to $\tilde{\mathcal{K}}$ (see Figure 7.4(b)), such that replacing $\boldsymbol{Q}$ with $\tilde{\mathcal{K}} \star \boldsymbol{K}_1$ in (7.13) yields

$$\boldsymbol{K}(\alpha_{\text{gs}}(\tilde{\mathcal{K}} \star \boldsymbol{K}_1)) = \mathcal{K} \star (\alpha_{\text{gs}}\tilde{\mathcal{K}} \star \boldsymbol{K}_1) \ . \tag{7.14}$$



(a) (b)

FIGURE 7.4: *(a) shows the standard gain scheduling strategy with an observer-based controller $\mathcal{K}$ interconnected with a scheduler $\alpha_{gs}\boldsymbol{Q}$.*
(b) shows the design of the total gain scheduling controller, where $\boldsymbol{Q}$ is defined as $\tilde{\mathcal{K}} \star \boldsymbol{K}_1$, which yields the total controller as $\boldsymbol{K}(\alpha_{gs}\boldsymbol{Q}) = \mathcal{K} \star (\alpha_{gs}\tilde{\mathcal{K}} \star \boldsymbol{K}_1)$.

Based on the structure of (7.11) $\tilde{\mathcal{K}}$ is in Appendix D determined as:

$$\tilde{\mathcal{K}} = \left[\begin{array}{cc|cc} \boldsymbol{A}_0 & \boldsymbol{0} & \boldsymbol{F}_0 & \boldsymbol{B}_0 \\ \boldsymbol{L}_{I0}\boldsymbol{C}_0 - r\boldsymbol{L}_0 & -r\boldsymbol{I} & -\boldsymbol{L}_{I0} & r\boldsymbol{I} \\ \hline -\boldsymbol{L}_0 & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{C}_0 & \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} \end{array}\right] , \tag{7.15}$$

such that the following interconnection

$$\mathcal{K} \star \tilde{\mathcal{K}} = \left[\begin{array}{cccc|cc} \boldsymbol{A}_0 - \boldsymbol{B}_0\boldsymbol{L}_0 - \boldsymbol{F}_0\boldsymbol{C}_0 & \boldsymbol{B}_0 & \boldsymbol{B}_0\boldsymbol{L}_0 & -\boldsymbol{B}_0 & \boldsymbol{F}_0 & \boldsymbol{B}_0 \\ \boldsymbol{0} & \boldsymbol{0} & r\boldsymbol{L}_0 & -r\boldsymbol{I} & \boldsymbol{L}_{I0} & r\boldsymbol{I} \\ -\boldsymbol{F}_0\boldsymbol{C}_0 & \boldsymbol{0} & \boldsymbol{A}_0 & \boldsymbol{0} & \boldsymbol{F}_0 & \boldsymbol{B}_0 \\ -\boldsymbol{L}_{I0}\boldsymbol{C}_0 & \boldsymbol{0} & \boldsymbol{L}_{I0}\boldsymbol{C}_0 + r\boldsymbol{L}_0 & -r\boldsymbol{I} & \boldsymbol{L}_{I0} & r\boldsymbol{I} \\ \hline -\boldsymbol{L}_0 & \boldsymbol{I} & \boldsymbol{L}_0 & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{C}_0 & \boldsymbol{0} & \boldsymbol{C}_0 & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \end{array}\right] \tag{7.16}$$

can be made, where it can be seen that the interconnection $\mathcal{K} \star \tilde{\mathcal{K}}$ truly is an identity system, as $\hat{\boldsymbol{\xi}}_0 = \tilde{\boldsymbol{\xi}}$ results in:

$$\boldsymbol{u} = \boldsymbol{u}_1 \tag{7.17}$$

$$\boldsymbol{e}_1 = \boldsymbol{e} \ , \tag{7.18}$$

which means that the control signal $\boldsymbol{u}_1$ from the second observer-based controller $\boldsymbol{K}_1$ will be the only control signal affecting the controlled system. Likewise, the error $\boldsymbol{e}_1$ provided to controller $\boldsymbol{K}_1$ is equal to the actual error of the system $\boldsymbol{e}$.

Furthermore it can be seen that when the augmented state vectors are identical they will stay that way as long as $\alpha_{\text{gs}} = 1$:

$$\dot{\hat{\boldsymbol{x}}}_0 = (\boldsymbol{A}_0 - \boldsymbol{F}_0\boldsymbol{C})\hat{\boldsymbol{x}}_0 + \boldsymbol{F}_0\boldsymbol{e} + \boldsymbol{B}_0\boldsymbol{u}_1 \tag{7.19}$$

$$\dot{\boldsymbol{x}}_{I0} = r\boldsymbol{L}_0\tilde{\boldsymbol{x}} - r\tilde{\boldsymbol{x}}_I + \boldsymbol{L}_{I0}\boldsymbol{e} + r\boldsymbol{u}_1 \tag{7.20}$$

$$\dot{\tilde{\boldsymbol{x}}} = -\boldsymbol{F}_0\boldsymbol{C}_0\hat{\boldsymbol{x}}_0 + \boldsymbol{A}_0\tilde{\boldsymbol{x}} + \boldsymbol{F}_0\boldsymbol{e} + \boldsymbol{B}_0\boldsymbol{u}_1 \tag{7.21}$$

$$\dot{\tilde{\boldsymbol{x}}}_I = r\boldsymbol{L}_0\tilde{\boldsymbol{x}} - r\tilde{\boldsymbol{x}}_I + \boldsymbol{L}_{I0}\boldsymbol{e} + r\boldsymbol{u}_1 \tag{7.22}$$

The structure of the second observer based controller $\boldsymbol{K}_1$ is the same as given in (7.12) but designed in a new operating point. $\boldsymbol{K}_1$ is then given as

$$\boldsymbol{K}_1 = \left[\begin{array}{cc|c} \boldsymbol{A}_1 - \boldsymbol{B}_1\boldsymbol{L}_1 - \boldsymbol{F}_1\boldsymbol{C}_y & \boldsymbol{B}_1 & \boldsymbol{F}_1 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L}_{I1} \\ \hline -\boldsymbol{L}_1 & \boldsymbol{I} & \boldsymbol{0} \end{array}\right] , \tag{7.23}$$

where $\boldsymbol{K}_1$ has the augmented state vector $\hat{\boldsymbol{\xi}}_1 = [\ \hat{\boldsymbol{x}}_1^T \quad \hat{\boldsymbol{x}}_{I1}^T\ ]^T$.

7.2.1 Design of observer gains $\boldsymbol{F}_0$ and $\boldsymbol{F}_1$

Having determined the structure of the observer-based gain scheduler and the LQR controllers the observer feedback gain matrices for the two operating points are the next to be designed. Each of the feedback gains is calculated using the MATLAB© function `lqed()`, which calculates the Kalman feedback gain matrix $\boldsymbol{F}$ when given the system matrix $\boldsymbol{A}$, process noise matrix $\boldsymbol{G}$, output matrix $\boldsymbol{C}$, observer estimat covariance matrix $\boldsymbol{Q}$, measurement covariance matrix $\boldsymbol{R}$ and the desired sampling time T_s as inputs. $\boldsymbol{A}$ is known from Appendix C for both operating points. The estimated states of the nonlinear model are the observer measurements associated with the covariance matrix $\boldsymbol{R}$. It is assessed that the measurements are reliable based on the tests performed for the LQR controllers in Chapter 6, and therefore the covariance matrix $\boldsymbol{R}$ is set small relative to $\boldsymbol{Q}$. By simulations with the nonlinear model in open loop and trial-and-error tuning the following matrices are determined:

$$\boldsymbol{R} = 0.00001\boldsymbol{I} \quad \wedge \quad \boldsymbol{Q} = 0.1\boldsymbol{I} \quad \wedge \quad \boldsymbol{G} = \boldsymbol{I} \tag{7.24}$$

The two calculated observer gains $\boldsymbol{F}_0$ and $\boldsymbol{F}_1$ can be found in Appendix E). The designed observer gain matrices can only be used for observer based gain scheduling approach if the observer gains are able to stabilize the system $\boldsymbol{A}$ such that $\boldsymbol{A} + \boldsymbol{F}\boldsymbol{C}$ (or $\boldsymbol{A} - \boldsymbol{F}\boldsymbol{C}$) is *Hurwitz*, which means that all eigenvalues must have strictly negative real parts (in the discrete case the eigenvalues must be within the unit circle)(Bendtsen et al. [2005]). By using the MATLAB© function `eig()` the eigenvalues for the two stable systems $\boldsymbol{A}_0 - \boldsymbol{F}_0\boldsymbol{C}_0$ and $\boldsymbol{A}_1 - \boldsymbol{F}_1\boldsymbol{C}_1$ are:

$$\texttt{eig}(\boldsymbol{A}_0 - \boldsymbol{F}_0\boldsymbol{C}_0) =$$

$$\Big[-14.7005 \quad -8.2241 \quad -5.4541 \quad -1.8940 \quad -0.9901 - 0.4069i \quad -0.9901 + 0.4069i$$

$$-1.0050 - 0.3886i \quad -1.0050 + 0.3886i \quad -0.9975 \quad -0.9975 \quad -0.9975 \quad -0.9975 \Big]^T$$

$$\texttt{eig}(\boldsymbol{A}_1 - \boldsymbol{F}_1\boldsymbol{C}_1) =$$

$$\Big[-15.2053 \quad -8.2385 \quad -3.5084 - 2.5335i \quad -3.5084 + 2.5335i \quad -2.9647$$

$$-1.0141 - 0.3440i \quad -1.0141 + 0.3440i \quad -1.0726 \quad -0.9940 - 0.0550i$$

$$-0.9940 + 0.0550i \quad -0.9971 - 0.0006i \quad -0.9971 + 0.0006i \Big]^T$$

As it can be seen above all eigenvalues for the two observers have strictly negative real parts, which shows that both systems are Hurwitz and therefore usable for observer-based gain scheduling.

7.2.2 Design of α_{gs} function

The scheduling variable $\alpha_{\text{gs}} \in [0;1]$ is modeled as a piecewise linear function of the forward flight velocity ${}^b\dot{x}$ as illustrated in Figure 7.5. As can be seen small values of α_{gs} exists for ${}^b\dot{x} < 4\,\frac{\text{m}}{\text{s}}$, and large values of α_{gs} (values close to one) exists for ${}^b\dot{x} > 6\,\frac{\text{m}}{\text{s}}$. In that way $\boldsymbol{K}_0$ obtains the most influence for ${}^b\dot{x} < 4\,\frac{\text{m}}{\text{s}}$ and $\boldsymbol{K}_1$ gets the most influence for ${}^b\dot{x} > 6\,\frac{\text{m}}{\text{s}}$. At a forward flight velocity of ${}^b\dot{x} = 5\,\frac{\text{m}}{\text{s}}$ equal influence of the two controllers is obtained.

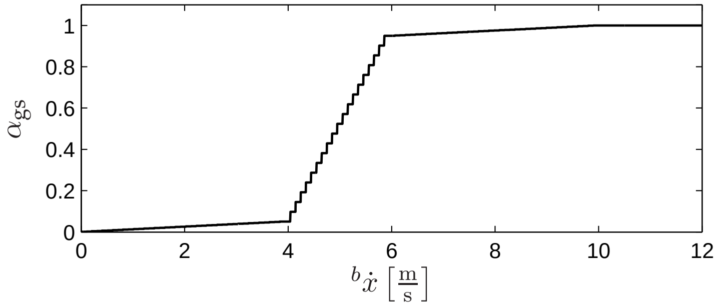


FIGURE 7.5: *Illustration of the chosen α_{gs} function.*

Having designed the α_{gs} function, the design of the observer-based gain scheduling controller is concluded. In the following section the functionality and performance of the controller will be evaluated.

7.3 Gain scheduling control test

In Section 7.2 the algebraic calculations showed that the interconnected system $\mathcal{K}\star\tilde{\mathcal{K}}$ is an identity system when $\alpha_{\text{gs}} = 1$. To determine whether the scheduler can actually be used for control of the nonlinear model it is implemented and tested with respect to the change of controller, which means that the gain scheduling system must be tested for the properties of $\mathcal{K} \star \tilde{\mathcal{K}}$. The property is tested by comparing the two state vectors $\hat{\boldsymbol{x}}_0$ and $\tilde{\boldsymbol{x}}$ from $\mathcal{K}$ and $\tilde{\mathcal{K}}$ respectively. If the interconnection between the two are indeed an identity system then $\hat{\boldsymbol{x}}_0 - \tilde{\boldsymbol{x}} \rightarrow \mathbf{0}|_{\alpha_{\text{gs}}=1}$.

The test of the gain scheduling system is performed by implementing the α_{gs} function as a ramp with a slope of 0.1 and saturation at $\alpha_{\text{gs}} = 1$. At the time zero $\alpha_{\text{gs}} = 0$ and after 10 seconds $\alpha_{\text{gs}} = 1$ is obtained. Because the scheduling system is initialized with $\alpha_{\text{gs}} = 0$ the states observed by $\mathcal{K}$ and $\tilde{\mathcal{K}}$ will be different. Therefore, it is expected that when the ramp function becomes 1 the scheduling system will make $\hat{\boldsymbol{x}}_0 \rightarrow \tilde{\boldsymbol{x}}$ for increasing time $t > 10\,\text{s}$. The result of the test of the scheduler can be seen in Figure 7.6. It is observed, that after time $t = 10\,\text{s}$ all observer errors converges to zero ($\hat{\boldsymbol{x}}_0 - \tilde{\boldsymbol{x}} \rightarrow 0$). It is therefore concluded that the scheduler is performing as intended with the ability to switch between the two controllers.

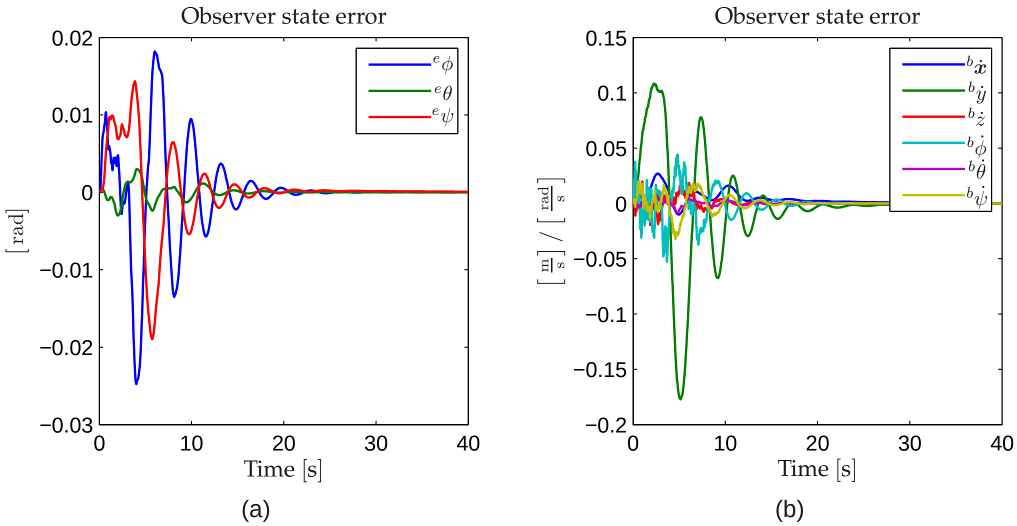


FIGURE 7.6: *Graphs showing the observer state error $\hat{\boldsymbol{x}}_0 - \tilde{\boldsymbol{x}}$ for α_{gs} going from 0 to 1. (a) shows the error in the system angles given in the EF. (b) shows the error in the system velocities given in the BF.*

Having determined that the scheduler is able to switch between the two LQR controllers the total observer-based gain scheduling control system can be validated using the same approach as with the classic and LQR controllers, by initializing the UAV with offsets in six of the control states. The offset values are the same as for the tests of the classic and the LQR controller (see Table 4.1 on page 22). However, it is desired to perform the test with a mixture of $\boldsymbol{K}_0$ and $\boldsymbol{K}_1$ such that both controllers are equally used. This means that an $\alpha_{\text{gs}} = 0.5$ resulting in a forward flight velocity of $5\,\frac{\mathrm{m}}{\mathrm{s}}$ must be maintained. The offset test for the observer-based gain scheduling control system can be seen in Figure 7.7.

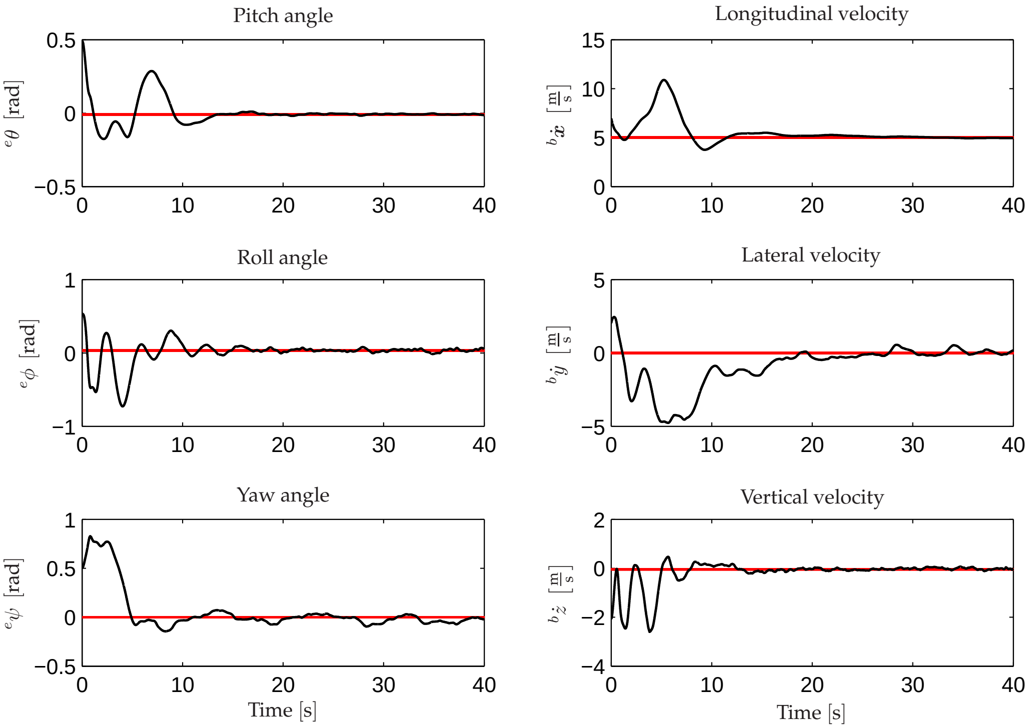


FIGURE 7.7: *Simulation result of the observer-based gain scheduling controllers ability to stabilize the UAV at $5\,\frac{\mathrm{m}}{\mathrm{s}}$ when given an offset in six of the controlled states.*

It is observed that rather large variations for all states during the first $10\,\mathrm{s}$ exists. These are explained by the estimator problem mentioned in Section 6.4 (applies to the Sensor fusion and estimation block presented in Section 2.2) causing the estimated state ${}^b\dot{x}_{\text{est}}$ to start at $0\,\frac{\mathrm{m}}{\mathrm{s}}$. Overall, at time $t = 0\,\mathrm{s}$, the real state is $7\,\frac{\mathrm{m}}{\mathrm{s}}$, the estimated state is ${}^b\dot{x}_{\text{est}} = 0\,\frac{\mathrm{m}}{\mathrm{s}}$ and the reference is $r_{{}^b\dot{x}} = 5\,\frac{\mathrm{m}}{\mathrm{s}}$. The error based on the estimated state seen from the controller is therefore $-5\,\frac{\mathrm{m}}{\mathrm{s}}$, where the actual error is $2\,\frac{\mathrm{m}}{\mathrm{s}}$, causing the controller to bring the UAV further away from its reference. Despite the fact that ${}^b\dot{x}_{\text{est}}$ becomes equal to ${}^b\dot{x}$ after approximately $2\,\mathrm{s}$, the large variations continues the first $10\,\mathrm{s}$ as a result of the integral action in the controller accumulating the errors. The effort of the controller trying to stabilize the UAV based on a wrong state estimate propagates to the other states as well due to cross couplings in the model. However, the controller stabilizes the UAV at the references afterwards.

For the final test of gain scheduling controller, an acceleration from hover to forward flight of $10\,\frac{\mathrm{m}}{\mathrm{s}}$ is simulated with the nonlinear model in SIMULINK. The UAV is initialized in hover, and four steps of each $2.5\,\frac{\mathrm{m}}{\mathrm{s}}$ at the times $10, 20, 30$ and $40\,\mathrm{s}$ are used to produce the forward flight reference $r_{^b\dot{x}}$. The longitudinal velocity and the pitch angle from the test simulation is shown in Figure 7.8. It is observed that $^b\dot{x}$ has a small overshoot at the first two steps and no overshoot at the last two steps, which indicates that different controllers are used as intended. Overall the gain scheduling control system follows the reference without steady state error satisfying for use in the overall control system.

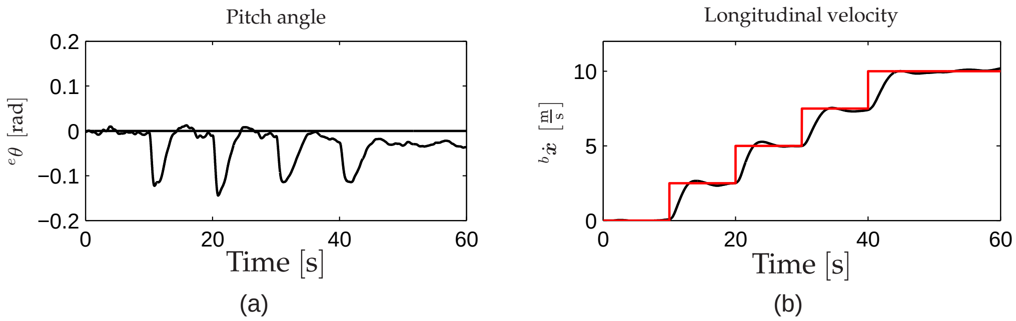


FIGURE 7.8: *Illustration of; (a) the pitch angle (b) the forward flight velocity (the red line indicates the reference) in a simulation of the gain scheduling controller in closed loop with the nonlinear model.*

Having designed and tested the gain scheduling controller of the overall control system, the last element of the control system, the supervisor, can be designed, which is done in the following chapter.

8 Supervisory control

This chapter describes the design of the supervisory controller used for providing references to the low level linear controllers. Different algorithms used for calculating the references based on simple rules on how to handle the approach to and turning at waypoints are described. A number of parameters associated with thresholds for the different decisions are determined as described last in the chapter.

8.1 Overview

The purpose of the supervisor is to provide position control for the control system. Based on position and heading it must be able to generate references for the LQR and gain scheduling control units, such that the UAV is able to complete level 1 in the IARC as described in section 3.1. These references are created by rules on how to handle all possible situations given the actual position and heading of the UAV compared to the target waypoint (the next waypoint to be reached). The following in- and outputs are thus identified:

- **Input:** Level 1 specification:
  ${}^e\boldsymbol{\Xi}_{\text{lvl1}} = \left\{ \begin{bmatrix} {}^ex_{\text{wp},1} & {}^ey_{\text{wp},1} & {}^ez_{\text{wp},1} \end{bmatrix}, \ldots, \begin{bmatrix} {}^ex_{\text{wp},n} & {}^ey_{\text{wp},n} & {}^ez_{\text{wp},n} \end{bmatrix} \right\}$
- **Input:** Feedback control signals: ${}^e\boldsymbol{\Xi} = \begin{bmatrix} {}^ex & {}^ey & {}^ez \end{bmatrix}$ and ${}^e\psi$
- **Output:** Reference signals: $\boldsymbol{o}_r = \begin{bmatrix} r_{{}^e\phi} & r_{{}^e\theta} & r_{{}^e\psi} & r_{{}^e\dot{x}} & r_{{}^e\dot{y}} & r_{{}^e\dot{z}} & r_{{}^b\dot{\phi}} & r_{{}^b\dot{\theta}} & r_{{}^b\dot{\psi}} \end{bmatrix}$

In the IARC the waypoints are given as $\begin{bmatrix} {}^ex_{\text{wp}} & {}^ey_{\text{wp}} & {}^ez_{\text{wp}} \end{bmatrix}$, where the ${}^ez_{\text{wp}}$ coordinate is included such that the UAV can be controlled to maintain a certain altitude.

The last feedback signal ${}^e\psi$ is used in flight close to a waypoint if a *hover turn* must be performed, which means that the UAV stops above a waypoint, then turns towards the next waypoint and finally continues towards the next waypoint.

Before the design of control rules the strategy of the supervisor is described.

8.2 Supervisor strategy

Given a level 1 specification:

$$ {}^e\boldsymbol{\Xi}_{\text{lvl1}} = \left\{ \begin{bmatrix} {}^ex_{\text{wp1}} & {}^ey_{\text{wp1}} & {}^ez_{\text{wp1}} \end{bmatrix}, \begin{bmatrix} {}^ex_{\text{wp2}} & {}^ey_{\text{wp2}} & {}^ez_{\text{wp2}} \end{bmatrix}, \begin{bmatrix} {}^ex_{\text{wp3}} & {}^ey_{\text{wp3}} & {}^ez_{\text{wp3}} \end{bmatrix} \right\} $$

and assuming that the UAV is initialized in hover with the altitude of the first waypoint, a simple control strategy repeated for all waypoints can be described as:

1. Turn towards target waypoint.

2. Fly to target waypoint.

3. Update target waypoint.

4. Repeat for n waypoints.

The challenge lies in identifying appropriate rate limitations for the calculated output references in order to obtain a steady flight. The rate limitations are especially important when passing a waypoint and thereby updating the target waypoint, which, without rate limitation, would induce unwanted reference changes in terms of steps. In addition, a suitable turn method must be identified for each waypoint to further ensure a steady flight, but also in some situations allow the UAV to pass waypoints without stopping to reduce the total flight time in level 1.

8.3 Control design

The control strategy described above is illustrated by the flowchart in Figure 8.1. In the following sections the different decision and action blocks will be elaborated on. A number of constants will for the sake of clarity be referred to only by symbols. The exact sizes of these are determined as described in section 8.4 on page 82.

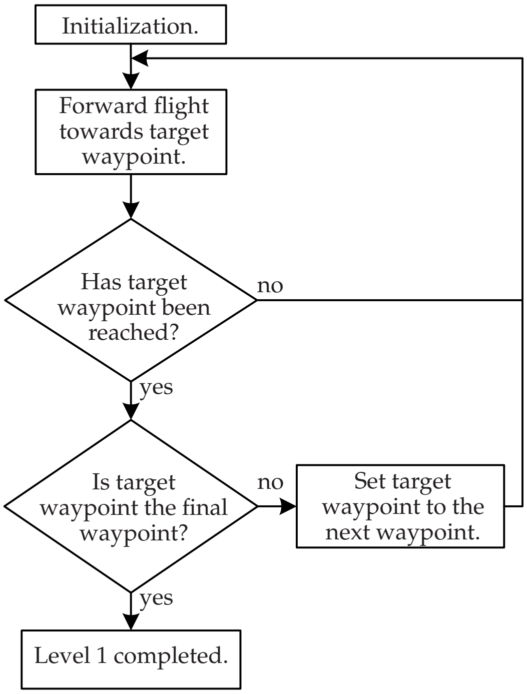

Initialization.
Forward flight towards target waypoint.
Has target waypoint been reached?
no
yes
Is target waypoint the final waypoint?
no
Set target waypoint to the next waypoint.
yes
Level 1 completed.

FIGURE 8.1: *Illustration of the overall supervisor control strategy. The UAV is initialized in hover with the altitude equal to that of the first waypoint with a heading* ${}^{e}\psi = 0\,\mathrm{rad}$.

8.3.1 Initialization

As depicted in Figure 8.2 the control strategy is based on two paths yielding two possible turning methods; A *Hover turn* where path (a) is followed, and a *Forward flight turn* where path (b) is followed.

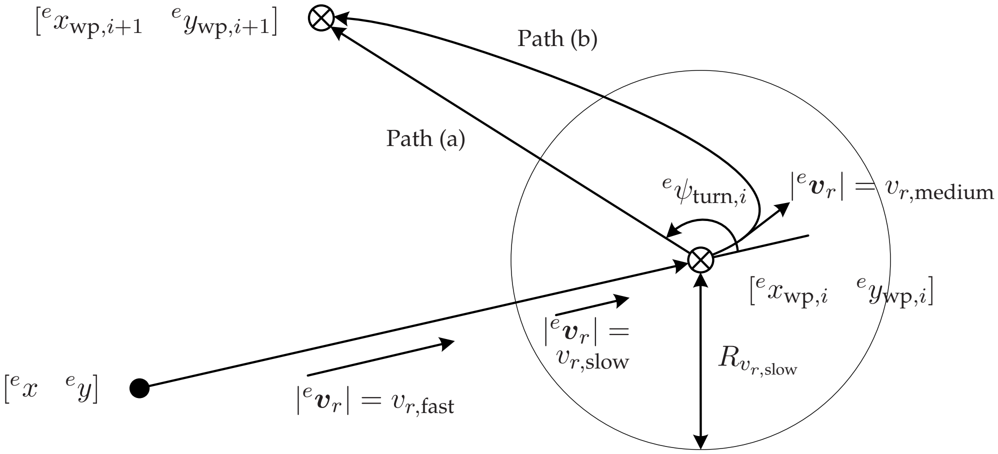

$[{}^e x_{\text{wp},i+1} \quad {}^e y_{\text{wp},i+1}]$
Path (b)
Path (a)
${}^e\psi_{\text{turn},i}$
$|{}^e\boldsymbol{v}_r| = v_{r,\text{medium}}$
$[{}^e x_{\text{wp},i} \quad {}^e y_{\text{wp},i}]$
$|{}^e\boldsymbol{v}_r| = v_{r,\text{slow}}$
$R_{v_{r,\text{slow}}}$
$[{}^e x \quad {}^e y]$
$|{}^e\boldsymbol{v}_r| = v_{r,\text{fast}}$

FIGURE 8.2: *Illustration of factors and approach associated with calculation of the velocity reference ${}^e\boldsymbol{v}_r$ and the identification of turn method. Here shown for the waypoint $[{}^e x_{wp,i} \quad {}^e y_{wp,i}]$.*

Hover turn:

A hover turn consists of the following actions:

1. Decelerate towards waypoint ${}^e\boldsymbol{\Xi}_{\text{wp},i}$ until it is reached.
2. Hover and turn towards ${}^e\boldsymbol{\Xi}_{\text{wp},i+1}$.
3. Accelerate towards ${}^e\boldsymbol{\Xi}_{\text{wp},i+1}$ following path (a).

In the hover approach a velocity reference ${}^e\boldsymbol{v}_r$ consisting of a longitudinal ($r_{{}^e\dot{x}}$) and a lateral ($r_{{}^e\dot{y}}$) component is set to the size $v_{r,\text{slow}}$, when a certain distance $R_{v_{r,\text{slow}}}$ (indicated by the circle) has been reached. Due to the rate limit on the references, the shift from $v_{r,\text{fast}}$ to $v_{r,\text{slow}}$ results in the deceleration referred to above. As ${}^e\boldsymbol{v}_r$ is mapped from the EF to the BF and then provided to the LQR controllers, it is expected that the lateral reference component $r_{{}^b\dot{y}}$ in the BF is close to zero when approaching the waypoint having a velocity reference $v_{r,\text{slow}}$. Assuming that the LQR controller is capable of keeping the controlled states at the references, no sideways movement of the UAV can be expected. The *hover and turn action* in point 2. above is performed by keeping ${}^e\boldsymbol{v}_r$ at zero, from the time a waypoint is reached, until the heading of the UAV becomes equal to the heading reference.

Forward flight turn:

A forward flight turn consists of the following actions:

1. Decelerate at $R_{v_{r,\text{slow}}}$ to $v_{r,\text{medium}}$ towards waypoint ${}^e\boldsymbol{\Xi}_{\text{wp},i}$ until it is reached.
2. Turning with $v_{r,\text{medium}}$ until the direction of ${}^e\boldsymbol{\Xi}_{\text{wp},i+1}$ is reached.
3. Accelerate to $v_{r,\text{fast}}$ towards ${}^e\boldsymbol{\Xi}_{\text{wp},i+1}$ following path (b).

The idea with the forward flight turn is to avoid a deceleration to $v_{r,\text{slow}}$ and hereby optimize the time consumption while performing a turn. The helicopter makes the deceleration at $R_{v_{r,\text{slow}}}$ to $v_{r,medium}$ and keep the heading until ${}^e\boldsymbol{\Xi}_{\text{wp},i}$ is reached. At the waypoint a new velocity reference vector ${}^e\boldsymbol{v}_r$ pointing in the direction of ${}^e\boldsymbol{\Xi}_{\text{wp},i+1}$ is provided to the controllers immediately. If the next waypoint do not have the same heading as the previous waypoint, there can occur some sideways flight, hereof a velocity in ${}^b\dot{y}$. The UAV will keep on turning until the heading ${}^e\psi$ has reached the reference $r_{{}^e\psi}$, and hereafter the helicopter starts to accelerate to the maximum velocity of the given controller.

Choosing turn method:

The supervisor must decide once for each waypoint to perform either a hover turn or a forward flight turn. This decision will be based on:

1. the distance $\ell_{\text{wp},i\rightarrow i+1}$ between ${}^e\boldsymbol{\Xi}_{\text{wp},i}$ and ${}^e\boldsymbol{\Xi}_{\text{wp},i+1}$ and,
2. the turn angle ${}^e\psi_{turn,i}$ measured between the continuation of a virtual straight line between ${}^e\boldsymbol{\Xi}_{\text{wp},i-1}$ and ${}^e\boldsymbol{\Xi}_{\text{wp},i}$ and another straight line between ${}^e\boldsymbol{\Xi}_{\text{wp},i}$ and ${}^e\boldsymbol{\Xi}_{\text{wp},i+1}$ according to Figure 8.2.

The idea is, that if the distance $\ell_{\text{wp},i\rightarrow i+1}$ is large enough and the turn angle ${}^e\psi_{\text{turn},i}$ is not too large, a forward flight turn is performed in order to reduce the total flight time of the level. Appropriate constant limits must therefore be determined by simulations, such that the following two statements can be tested by the supervisor:

$$\begin{aligned} &\textbf{if} \;\; \ell_{\text{wp},i\rightarrow i+1} < \ell_{\text{turn,hover}} \quad \textbf{then} \text{ perform hover turn} \\ &\textbf{else if} \;\; |{}^e\psi_{turn,i}| > {}^e\psi_{\text{turn,hover}} \quad \textbf{then} \text{ perform hover turn} \;\;, \end{aligned}$$

where $\ell_{\text{turn,hover}}$ is the limit on the distance and ${}^e\psi_{\text{turn,hover}}$ the limit on the turn angle.

8.3.2 Decision blocks

In the first decision block it must be determined whether the current waypoint ${}^e\boldsymbol{\Xi}_{\text{wp},i}$ has been reached. For this purpose a circle of radius $R_{v_{r,\text{hover}}}$ is defined in the xy-plane. When the UAV enters this circle the given waypoint has been reached.

The second decision keeps track of the target waypoint, and terminate when the last waypoint has been reached. If the waypoint is not the last, the target waypoint is set to the next, which is done in the successive action block.

8.3.3 Forward flight

The action block *Forward flight towards the target waypoint* handles the following tasks:

1. Calculate heading reference $r_{^e\psi}$ towards the target waypoint.
2. Calculate velocity references ${}^e\boldsymbol{v}_r = \begin{bmatrix} r_{^e\dot{x}} & r_{^e\dot{y}} \end{bmatrix}$.

In addition, the forward flight action block contains a control of the altitude of the UAV, which simply increases or decreases the velocity reference $r_{^e\dot{z}}$ when the difference between the desired altitude ${}^ez_{\text{wp},i}$ and the feedback control signal ${}^ez$ exceeds a certain threshold. Furthermore, it contains rate limit control on all the calculated references $\begin{bmatrix} r_{^e\psi} & r_{^e\dot{x}} & r_{^e\dot{y}} & r_{^e\dot{z}} \end{bmatrix}$. Due to the simplicity of these controls, they will not be described any further.

Calculation of heading reference

Assuming that the UAV has the best flying properties in forward direction, it is desired to keep the heading given by ${}^e\psi$ towards the target waypoint at all times. This is done by calculating $r_{^e\psi}$ continuously during flight, rather than once at each waypoint. As can be seen from Figure 8.3 this is easily done by the tangent calculation:

$$r_{^e\psi} = \frac{\pi}{2} - \arctan(\frac{{}^ex_{\text{wp},i} - {}^ex}{{}^ey_{\text{wp},i} - {}^ey}) \quad , \tag{8.1}$$

with the special situations:

$${}^ey_{\text{wp},i} - {}^ey = 0 \tag{8.2}$$

$${}^ey_{\text{wp},i} - {}^ey < 0 \tag{8.3}$$

If (8.2) is valid the calculation will fail, as dividing by zero is not possible. If (8.3) is valid the result will be wrong, because tangent only evaluates to positive values.

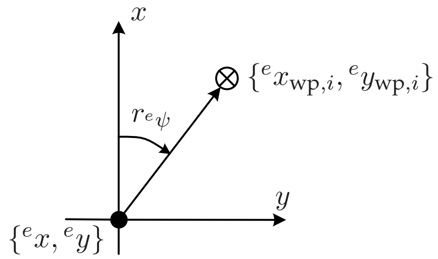


FIGURE 8.3: *Illustration of the frame used to calculate the heading reference $r_{^e\psi}$. The frame has the same orientation as the EF with an offset equal to the position of the UAV.*

Calculation of velocity references

When the size of the velocity reference $|{}^{e}\boldsymbol{v}_r|$ has been set to either $v_{r,\text{slow}}$ or $v_{r,\text{fast}}$, the velocity references $r_{^{e}\dot{x}}$ and $r_{^{e}\dot{y}}$ can be calculated (here for $v_{r,\text{slow}}$) as:

$$r_{^{e}\dot{x}} = \sin(\tfrac{\pi}{2} - r_{^{e}\psi}) \cdot v_{r,\text{slow}} \tag{8.4a}$$

$$r_{^{e}\dot{y}} = \cos(\tfrac{\pi}{2} - r_{^{e}\psi}) \cdot v_{r,\text{slow}} \tag{8.4b}$$

8.4 Parameter determination

To finish the design of the supervisor the different parameters have to be determined. The parameters determination will be done based on knowledge of the system and the rules associated with the IARC. As described in Section 4.1 on page 21 the UAV must pass a waypoint so a judge can see it from the ground by looking straight up and therefore $R_{v_{r,\text{hover}}}$ is chosen to $2\,\text{m}$. Furthermore, the velocity reference $v_{r,\text{slow}}$, used for a steady approach to a hover turn, is chosen to $1\,\frac{\text{m}}{\text{s}}$. Based on the determination of $v_{r,\text{slow}}$, the parameters $v_{r,\text{rate limit}}$, $v_{r,\text{fast}}$ and $R_{v_{r,\text{hover}}}$ for each control strategy can be determined by experimentation (see Appendix F). Then by having determined the different values of $v_{r,\text{fast}}$ for the control strategies, the medium velocity reference $v_{r,\text{medium}}$ is chosen such, that it is equal to half of the maximum velocity references. The different values of the determined supervisor parameters can be seen in Table 8.1. The two remaining parameters $\ell_{\text{turn,hover}}$ and ${}^{e}\psi_{\text{turn,hover}}$, which decides whether a hover turn must be performed or not, are determined by simulation tests. The simulations show that the UAV is capable of performing a $180°$ turn using the forward flight turn approach by decelerating to the medium velocity reference $v_{r,\text{medium}}$. Therefore, the implementation of the maximum turn angle ${}^{e}\psi_{\text{turn,hover}}$ is omitted from the supervisor. Having determined that a hover turn is only necessary to define based on the distance between two waypoints, the distance $\ell_{\text{turn,hover}}$ can be determined as the distance between two waypoints, involving a $180°$ turn, where a forward flight turn is as fast as a hover turn (see Figure

Parameter	Hover	Forward flight	Gain scheduling	Unit
$R_{v_{r,\text{hover}}}$	2	2	2	[m]
$R_{v_{r,\text{slow}}}$	37	60	62	[m]
$v_{r,\text{slow}}$	1	1	1	$\left[\frac{\text{m}}{\text{s}}\right]$
$v_{r,\text{fast}}$	11	15	16	$\left[\frac{\text{m}}{\text{s}}\right]$
$v_{r,\text{medium}}$	5.5	7.5	8	$\left[\frac{\text{m}}{\text{s}}\right]$
$v_{r,\text{rate limit}}$	± 5	± 5	± 5	$\left[\frac{\text{m}}{\text{s}^2}\right]$
$\ell_{\text{turn,hover}}$	100	150	100	[m]
${}^{e}\psi_{\text{turn,hover}}$	$\pm\pi$	$\pm\pi$	$\pm\pi$	[rad]

TABLE 8.1: *List of parameter values used by the supervisor together with each of the three controllers.*

8.4(a) for illustration). Based on test simulations it is determined that if the distance between two waypoints is greater than $100\,\mathrm{m}$ a forward flight turn is the most effective turning method using the hover or the gain scheduling controllers ($150\,\mathrm{m}$ in case of forward flight controller). Another example of the advantage of using forward flight turn is depicted in Figure 8.4(b). Here a $45°$ turn must be made with a distance of $300\,\mathrm{m}$ to the next waypoint. From the test it is observed that performing a forward flight turn is 5 seconds faster than performing a hover turn.

Having determined all of the necessary parameters for the supervisor it is possible to test the performance of the overall control system, which will be described in the following chapter.

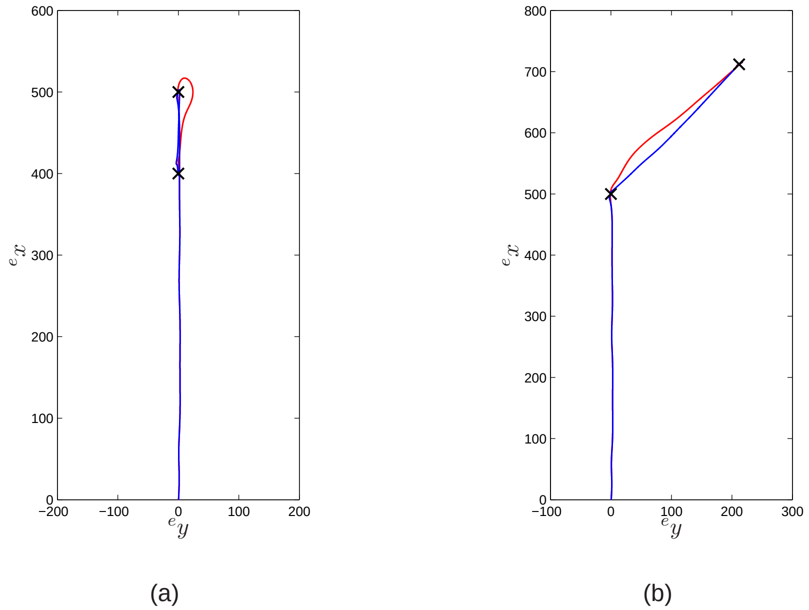


FIGURE 8.4: *Illustrations for determination of $\ell_{turn,hover}$ where the blue line is a hover turn and the red line in a forward flight turn - (a) $180°$ turn where both the completion times are $62\,\mathrm{s}$. (b) $45°$ turn where the completion times are $70\,\mathrm{s}$ and $65\,\mathrm{s}$ for the forward flight turn and hover turn, respectively.*

Part III

Control system test and conclusion

9
Control system test

This chapter describes the conclusive tests performed on the three control strategies; LQR hover control, LQR forward flight control and observer-based gain scheduling control. Three tests regarding different level 1 specifications for the IARC are performed for each control strategy, such that the most time efficient strategy can be identified. In addition the performance of the three control strategies is evaluated for the last level 1 specification.

9.1 Test specification

Starting point for the control system test is taken in the overall objective described in the introduction, hence the purpose is to identify the most efficient control strategy regarding completion time of level 1 of the IARC. The control strategies to be tested are; LQR hover control, LQR forward flight control and gain scheduling control. The three controllers are each tested together with the supervisor. To identify the best overall control strategy, it is desired to test the control system in different situations regarding the level 1 specification of the IARC, such that the best overall control strategy can be identified. The three different level 1 specifications ($\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$) are depicted in Figure 9.1. Each specification has a length of $3\,\mathrm{km}$ and contains a starting and end point, and up to 4 waypoints pursuant to the IARC rules.

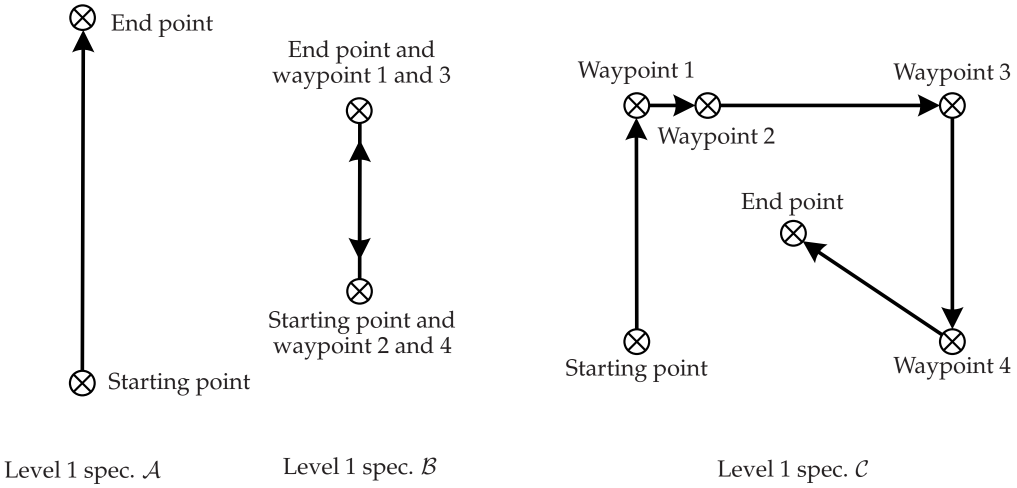

End point
Starting point
End point and waypoint 1 and 3
Starting point and waypoint 2 and 4
Waypoint 1
Waypoint 2
Waypoint 3
End point
Starting point
Waypoint 4
Level 1 spec. $\mathcal{A}$
Level 1 spec. $\mathcal{B}$
Level 1 spec. $\mathcal{C}$

FIGURE 9.1: *Illustration of the three level 1 specifications, all with a total flight length of $3\,\mathrm{km}$ used for performing the control system test. $\mathcal{A}$ has no waypoints, $\mathcal{B}$ has four waypoints placed such that a $180°$ turn must be performed at each waypoint, and $\mathcal{C}$ also has four waypoints placed such that several different turn types are obtained.*

Level 1 spec. $\mathcal{A}$: This specification is assumed to be the least time demanding of the three, as it simply consists of a starting and an end point placed $3\,\mathrm{km}$ from each other. The ability to accelerate and decelerate together with the maximum flight velocity are the decisive factors for identifying the best control strategy for $\mathcal{A}$.

Level 1 spec. $\mathcal{B}$: This specification is assumed to be the most time demanding specification, as it beside starting and end point consists of four waypoints placed such that a $180°$ turn has to be performed at each waypoint. The ability to accelerate, decelerate and turn will be the decisive factors in this test.

Level 1 spec. $\mathcal{C}$: The last specification also contains four waypoints, but the waypoints are placed such that different turn angles are obtained. Waypoint 1 and 2 are intentionally placed such that the supervisor is forced to decide for a hover turn at waypoint 1. Graphs of the six control states will also be presented for Level 1 spec. $\mathcal{C}$, and the performance of the three control strategies regarding the states will be discussed.

In the following sections the test results are discussed. To ease the reading the abbreviation LQR is left out when referring to the LQR hover and/or the LQR forward flight controller.

9.2 Flight performance in specification $\mathcal{A}$

As described above the ability to accelerate and decelerate and the maximum forward flight velocity are the decisive factors in $\mathcal{A}$. From Table 8.1 on page 82 it can be seen that the gain scheduling controller has the highest forward flight velocity reference $\boldsymbol{v}_{r,\text{fast}}$ and it is therefore expected that using the gain scheduling strategy to complete $\mathcal{A}$ takes less time than with the hover or the forward flight controller.

The resulting flight path in the xy-plane is depicted in Figure 9.2 where the blue, red and cyan line are the hover, forward flight and gain scheduling strategy respectively. It is observed that with the hover control strategy the UAV deviates more from the optimal path, indicated by the straight line, than with the other strategies. This is because the flight mainly consists of forward flight with maximum velocity, which means that the hover controller is far from the operating point for which it is developed causing it to drift in the ${}^{e}y$ direction. The flight route for the gain scheduling and the forward flight strategies are much alike. This is because the gain scheduling controller decouples the hover controller at high velocities, such that only the control signals from the forward flight controller are used in the observer construction.

The completion time for each control strategy is listed in Table 9.1. As expected the gain scheduling strategy has the lowest completion time.

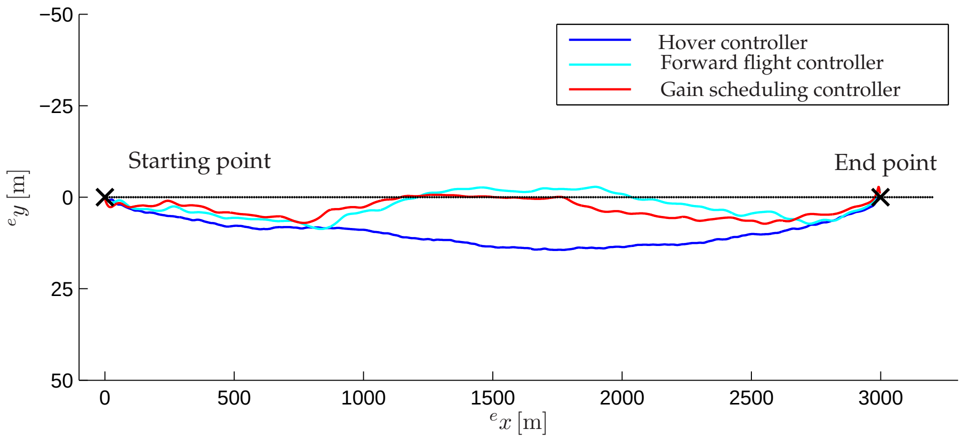


FIGURE 9.2: *Illustration of the flight paths from the simulation of $\mathcal{A}$. It is observed that the flight paths using forward flight and gain scheduling strategies are much alike, and that the hover strategy causes the UAV to drift with respect to the optimal path between starting and end point.*

Strategy	Hover	Forward flight	Gain scheduling
Completion time	287 s	219 s	204 s

TABLE 9.1: *Completion times for the three flight paths of the UAV in level 1 specification $\mathcal{A}$.*

9.3 Flight performance in specification $\mathcal{B}$

As mentioned earlier specification $\mathcal{B}$ consists of four waypoints placed on top of starting and end point (see Figure 9.1), which results in the flight paths being placed on top of each other as well. Consequently only two parts of the resulting flight path are presented (see Figure 9.3(a) and 9.3(b)). It is observed that the forward flight and the gain scheduling strategies, as was also the case in specification $\mathcal{A}$, results in similar flight paths. The same explanation as before applies to the flight in specification $\mathcal{B}$. Notice that forward flight turns with the velocity references $\boldsymbol{v}_{r,\text{medium}}$ listed in Table 8.1 are performed for all waypoints, as the distance between each waypoint is sufficient for the UAV to get back on track after each turn. Consequently the gain scheduling strategy primarily uses the forward flight controller.

From Figure 9.3(a) it is observed that the UAV turns right when using by the forward flight and the gain scheduling strategies and left for the hover strategy. The right turn is performed because the UAV approaches waypoint 1 from left and the supervisor therefore chooses to perform a right turn as it is the shortest (the opposite way for the hover strategy).

As given in Table 8.1, the distance $R_{\boldsymbol{v}_{r,\text{slow}}}$, indicating when a given controller must start decelerating towards a waypoint, varies between 37 m and 62 m, and as the distance between each

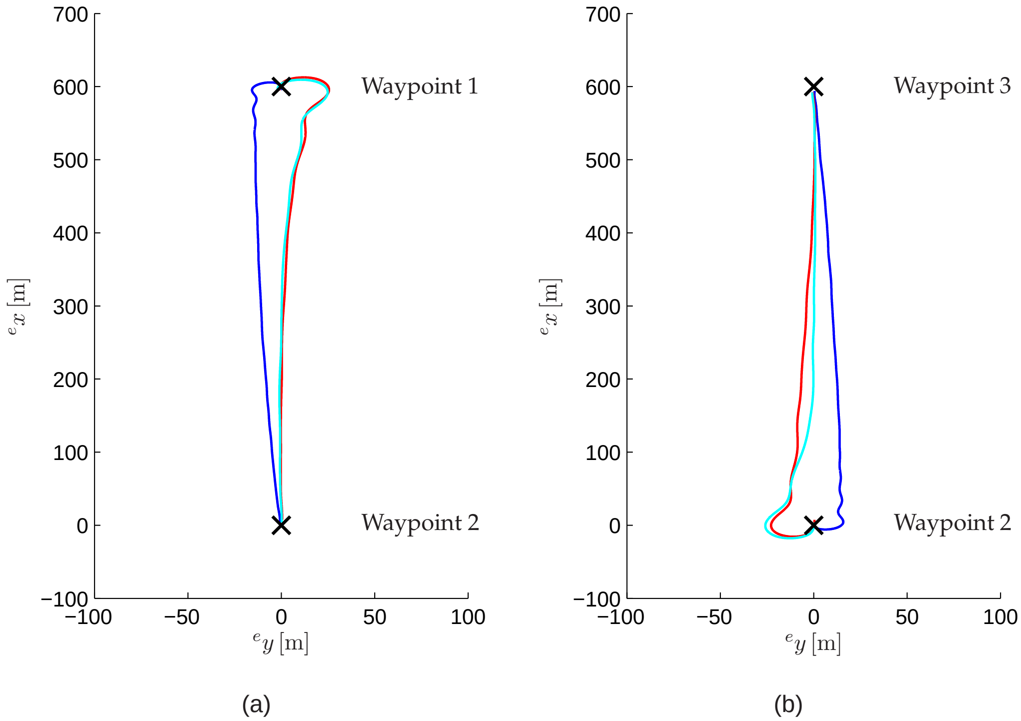


FIGURE 9.3: *The parts from waypoint 1 to 2 and waypoint 2 to 3 of the flight path from the simulations of specification $\mathcal{B}$. It is observed that the flight paths using forward flight (red) and gain scheduling (cyan) strategies are similar causing the UAV to make right turns at the waypoints, where left turns are performed for the hover strategy (blue).*

waypoint in specification $\mathcal{B}$ is $600\,\mathrm{m}$ the test flights primarily consists of flight with maximum velocity. The higher maximum velocity for the forward flight and the gain scheduling strategy, with respect to the hover strategy, therefore results in smaller completion times as listed in Table 9.2.

Looking at the split times for waypoint 1 and 2 for the gain scheduling controller, it can be shown that it takes approximately $4\,\mathrm{s}$ to perform a $180°$ turn. $1\,\mathrm{s}$ is subtracted from both time marks as the UAV initiates in hover, where the velocity when leaving waypoint 1 after completing the turn is approximately $5\,\frac{\mathrm{m}}{\mathrm{s}}$ yielding $t_{180} = (91 - 1) - 2 \cdot (44 - 1) = 4\,\mathrm{s}$. The same result of $4\,\mathrm{s}$ can obtained for the hover and the forward flight strategies, which indicates that the difference in the completion times follows from the ability of the strategies to accelerate and decelerate, and not from the ability to turn.

Strategy	Waypoint 1	Waypoint 2	Waypoint 3	Waypoint 4	End
Hover	59 s	121 s	182 s	244 s	311 s
Forward flight	47 s	97 s	148 s	199 s	253 s
Gain scheduling	44 s	91 s	139 s	187 s	238 s

TABLE 9.2: *Completion times for the UAV in specification $\mathcal{B}$ with split times at each waypoint.*

9.4 Flight performance in specification $\mathcal{C}$

The flight paths of the UAV flight in level 1 specification $\mathcal{C}$ are depicted in Figure 9.4 for hover (blue), forward flight (red) and gain scheduling (cyan) strategies respectively. The test results do not reveal any significantly differences between the flight paths for the three control strategies due to the scale, however, the completion times listed in Table 9.3 states, that using the gain scheduling control strategy is the most time efficient strategy. The test flight in specification $\mathcal{B}$ was assumed to be the most time demanding, however, the completion times for the test flights in specification $\mathcal{B}$ and $\mathcal{C}$ show that for the hover and forward flight strategies the flight in specification $\mathcal{C}$ is more time demanding. This is due to the hover turn present at waypoint 1. As expected the flight using

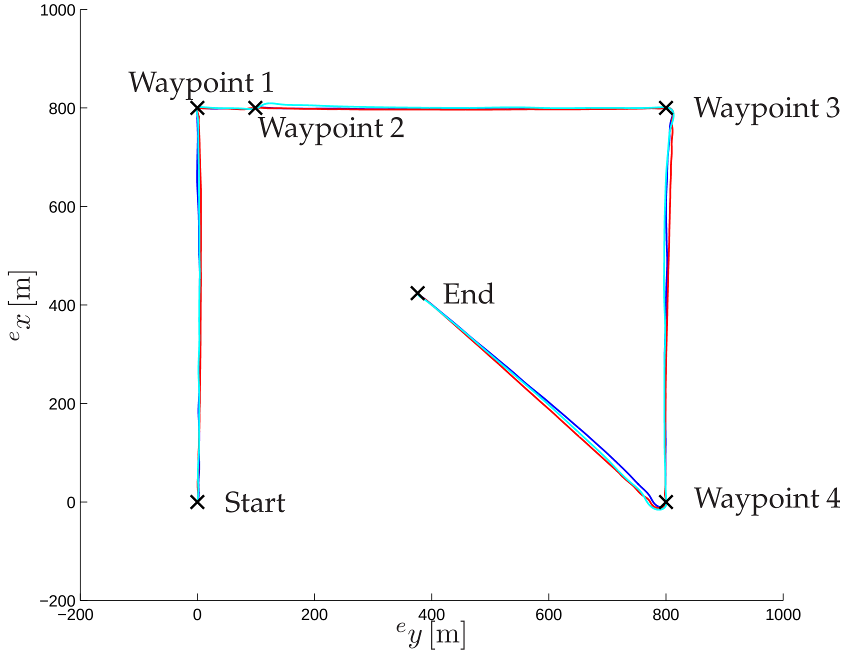


FIGURE 9.4: *Illustration of the flight paths from the simulations of specification $\mathcal{C}$ for hover (blue), forward flight (red) and gain scheduling (cyan) strategies respectively.*

Strategy	Hover control	Forward flight control	Gain scheduling control
Completion time	317 s	254 s	231 s

TABLE 9.3: *Completion times for the UAV for the simulation of specification $\mathcal{C}$.*

the gain scheduling strategy uses the least time, which again is related to the higher maximum velocity.

In the following the performance of the three control strategies are compared more thorough by investigating angle and translateral velocity states of the UAV. The six states are depicted in Figure 9.5, 9.6 and 9.7 for the hover, forward flight and gain scheduling strategy respectively.

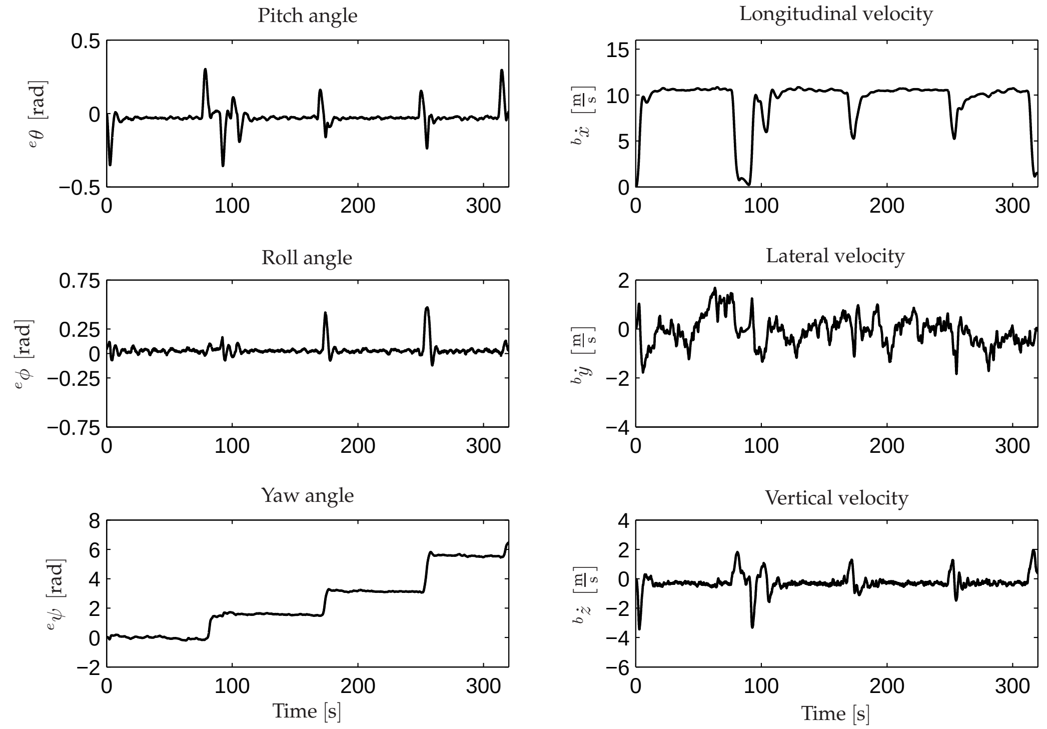


FIGURE 9.5: *Illustration of six of the control states found by simulation of the UAV flying trough specification $\mathcal{C}$ using the hover strategy .*

For all control strategies it is observed that the pitch angle becomes negative when accelerating in the longitudinal direction (and positive for deceleration), but with the forward flight strategy (see Figure 9.6) the UAV has larger peaks in the pitch angle, than with hover or gain scheduling strategy indicating a more aggressive controller. Furthermore, this indicates that the gain scheduling controller mixes the hover an forward flight controllers when ${}^b\dot{x}$ changes in the interval $[0; 10\,\frac{\mathrm{m}}{\mathrm{s}}]$ as intended, as it has smaller pitch angle variations than the forward flight strategy and greater variation than the hover strategy.

A similar observation applies to the roll angle graphs, where the forward flight strategy causes higher and more resonant variations than with the other strategies. The hover strategy affects the gain scheduling controller, such that a more steady control is obtained.

Overall it is concluded, from the whole control system test, that the gain scheduling control strategy is the most time efficient control strategy regarding flight in each of the three level 1 specifications discussed. In addition, the gain scheduling strategy is shown able to combine the two LQR controllers, in order benefit from the performance of the hover controller at low velocities and the forward flight controller at high velocities.

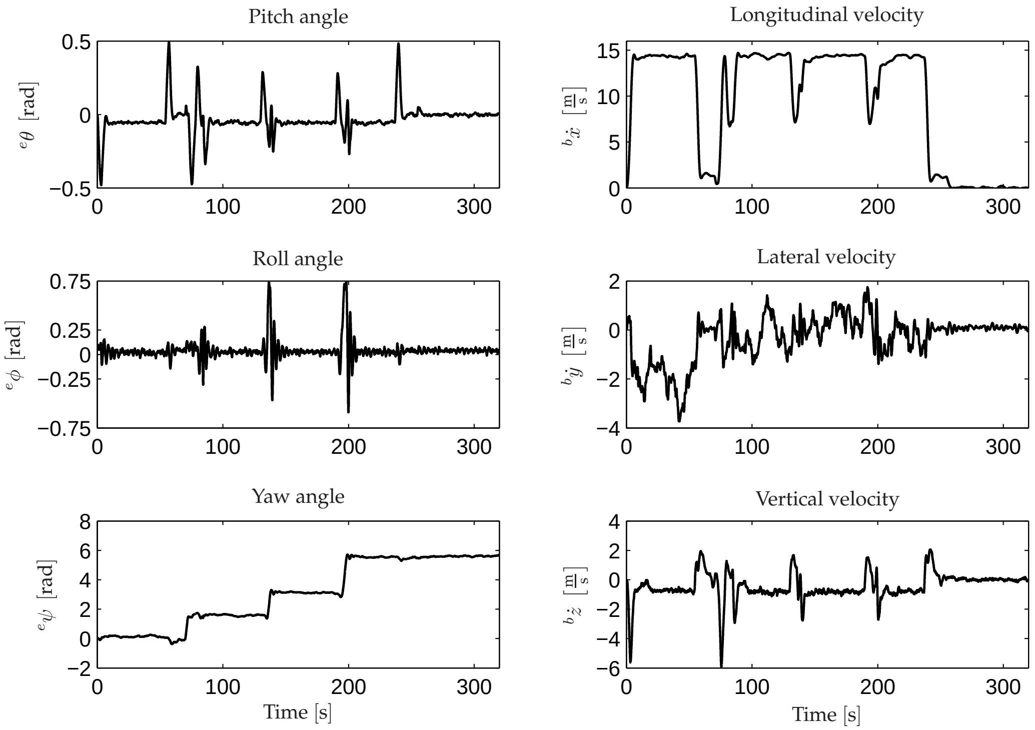


FIGURE 9.6: *Illustration of six of the control states found by simulation of the UAV flying trough specification C using the forward flight strategy.*

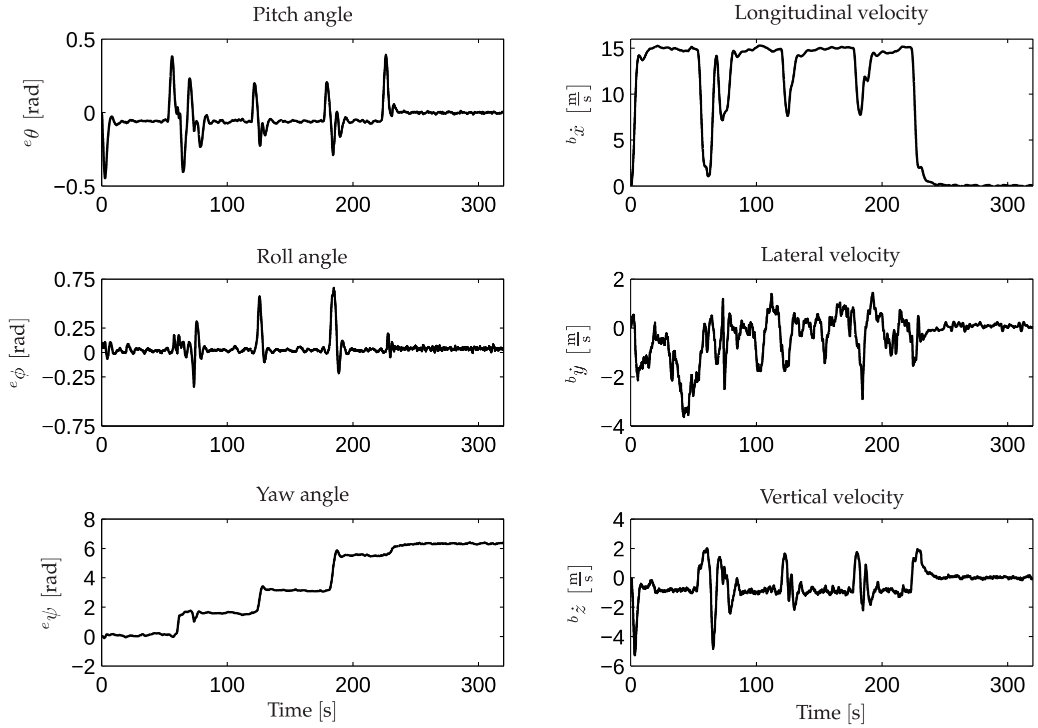


FIGURE 9.7: *Illustration of six of the control states found by simulation of the UAV flying trough specification $\mathcal{C}$ using the gain scheduling strategy.*

10 CONCLUSION

The overall goal of this master's thesis was to design control strategies such that the Bergen Industrial Twin helicopter is able to perform autonomous flight. The goal of the autonomous flight was to make the Unmanned Aerial Vehicle capable of competing in the International Aerial Robotics Competition and complete level 1 of this competition as fast as possible. Therefore, the overall objective for this thesis was determined to be:

> *Design, implementation and test of different control strategies enabling the UAV to autonomously complete level 1 in the IARC by employing hover and fast forward flight, such that the most efficient control strategy regarding level 1 completion time can be identified.*

To accommodate the overall objective various subobjectives were defined as described in Section 1.1. These objectives will be summarized in the following and concluded on separately followed by an overall conclusion of the achievements of this thesis.

Objective A - Classic linear control:

It was desired to design classical SISO controllers for the UAV to be able to compare this method to the optimal LQR method. The controllers were designed based on state-space matrices obtained by linearization of the nonlinear model in the hover operating point, and implemented as lead and lag compensators. Each of the designed controllers were tested separately and were all found to be able to stabilize their individual control state.

After all of the SISO controllers were designed the classic control system was tested, where the nonlinear model was initialized in hover with an offset applied to each of the controlled states. It could be concluded that the designed classic control system was able to stabilize the nonlinear model with respect to all of the controlled states.

Objective B - LQR hover control:

Besides the classical SISO control system it was desired to utilize more advanced control methods. For this purpose the MIMO control method optimal LQR control was used. The LQR hover control was, as the classic hover control, based on the linearized system matrices, and by identifying weight matrices for the controlled states. It was chosen to perform control on a total of nine states to be able to perform more stable control.

The main issue regarding the design was to determine the weight matrices used in the performance function. Initial values used in the weight matrices was determined by identifying state limitations of the nonlinear model in open loop. Using these determined weight matrices an initial test was performed to examine the performance of the LQR hover controller, where the test showed that the designed controller was not able to stabilize the UAV Therefore, the weight matrices were altered.

The final design of the LQR hover controller was tested on the nonlinear model initialized in hover with an offset applied to the controlled states as done for classic linear control. The test showed that the designed controller was able to stabilize all of the controlled states of the nonlinear model in a steady fashion and therefore assessed as performing satisfactory. Furthermore, it was concluded that the LQR hover controller performed more steady control than the classic linear controller, and was able to stabilize the UAV faster.

Objective C - LQR forward flight control:

A forward flight LQR controller was developed for an operating point in $10\,\frac{\text{m}}{\text{s}}$. The design of the this controller took starting point in the weight matrices determined for the LQR hover controller.

The LQR forward flight controller was tested as the LQR hover controller, however, the nonlinear model was initialized in the $10\,\frac{\text{m}}{\text{s}}$ operating point and then applied an offset in the controlled states. The test showed that the LQR forward flight controller was able to stabilize the nonlinear model in the operating point.

The tests performed for the forward flight controller showed the estimator to yield faulty estimates for the yaw angle and lateral velocity, which compromised the tests of the controller. However, due to the controllers ability to stabilize the UAV despite the faulty estimates, it was assessed that this fault was acceptable because it did not accumulate over time.

Objective D - Gain scheduling:

It was desired to optimize the change from hover to forward flight, why it was decided to design gain scheduling between the two LQR controllers. This was done by designing an observer-based gain scheduling controller to obtain bumpless transfer.

Since the two LQR controllers already were designed only the observer-based gain scheduler needed to be designed. For this purpose two observers were designed based on standard observer design theory.

To determine stability of the observer based gain scheduling controller a test was performed like for the LQR controllers. The nonlinear model was initialized in a $5\,\frac{\text{m}}{\text{s}}$ and then applied an offset in the controlled states. This operating point was chosen because both LQR controllers were equally used by the scheduler at this velocity.

The test showed that the observer based gain scheduling control system was able to stabilize the nonlinear model for this setting.

Objective E - Supervisory controller:

To obtain position and heading control a supervisory controller was developed. This high level controller was designed, such that it could calculate references for the linear controllers based on the position and heading of the UAV.

The supervisor was able to chose between two different waypoint turning methods; hover turn and forward flight turn, based on the waypoint position specification.

> The test of the supervisor was performed as part of the control system test, where it was concluded that it was able to handle various turn angles, and that it was able to minimize drift in the lateral direction at high velocities.

To identify the most time efficient control strategy three different control system tests were performed for each of the three control strategies; LQR hover, LQR forward flight and gain scheduling. From these tests it was identified that the gain scheduling strategy was the most time efficient of the three. The decisive factors were that the gain scheduling controller was able to benefit from the performance properties of both the LQR hover and forward flight controller. In addition, the gain scheduling controller was identified as having the highest maximum velocity.

10.1 Future work

The work done in this thesis qualifies as a basis for future work on control of the reconfigured Bergen Industrial Twin helicopter, as utilization of linear control showed good performance with respect to control of the nonlinear model. However, because the nonlinear model used in this thesis is of an older version, future controller designs must be based on the present nonlinear model.

Further optimization of the gain scheduling approach can advantageously be performed by adding several additional controllers in different operating points. In addition, a further analysis of the scheduling function α_{gs} could help optimize the benefit gained from each controller implemented in the gain scheduler. Implementing the α_{gs} scheduler function into the supervisor could also help optimize the use of the gain scheduling approach. This implementation could be used by the supervisor to chose between several additional rate limits as e.g. velocities.

A natural control approach to the Bergen Industrial Twin helicopter is to develop nonlinear control, which could be advantageous because of the complexity of the nonlinear model.

Acronyms

BF	Body-fixed reference Frame
EF	Earth-fixed reference Frame
IARC	International Aerial Robotics Competition
LQR	Linear Quadratic Regulator
MIMO	Multiple Input Multiple Output
SISO	Single Input Single Output
UAV	Unmanned Aerial Vehicle

Bibliography

AUVSI.
Association for unmanned vehicle systems international, 1972.
URL http://avdil.gtri.gatech.edu/AUVS/IARCLaunchPoint.html.
January 19th 2007.

J. D. Bendtsen, J. Stoustrup, and K. Trangbæk.
Bumpless transfer between observer-based gain scheduled controllers.
International Journal of Control, 78(7):491–504, 2005.

M. Bisgaard.
Ph.d student, 2007.
URL http://www.control.auc.dk/~bisgaard/phd.html.
November 6th 2006.

U. Hald, M. Hesselbæk, and M. Siegumfeldt.
Nonlinear modeling and optimal control of a miniature autonomous helicopter.
Master thesis, Aalborg University - Department of Control Engineering, 2006.
URL http://www.control.aau.dk/~alc/05gr937a_student_report.pdf.

B. R. Helicopters.
Wesite for bergen r/c helicopters, 2000.
URL http://www.bergenrc.com/IndustrialTwin.asp.
January 19th 2007.

J. Holmegaard, C. Jensen, and S. Lænner.
Development and navigation of an autonomous uav.
Master thesis, Aalborg University - Department of Control Engineering, 2006.
URL http://www.control.aau.dk/~alc/05gr937b_student_report.pdf.

P. K. Jensen, M. Y. Hundsdahl, H. Hansen, A. M. Iversen, L. S. Lantow, and J. H. Pedersen.
Development and application of a protocol for visualising simulations using a 3d engine.
Article, Aalborg University - Department of Control Engineering, 2006.

O. Sørensen.
Optimal regulering, 2. udgave.
Afdeling for proceskontrol, Aalborg Universitet, 1992.

K. Zhou, J. Doyle, and K. Glover.
Robust and Optimal Control.
Prentice-Hall, New Jersey, USA, 1996.
ISBN 0-13-456567-3.

Appendix A
Visualization program Gsim

This appendix describes how the visualization program Gsim *used in this thesis for visualization of the motion of the controlled UAV works. The program has especially been used when designing the LQR controllers in this thesis.*

A.1 Functionality of Gsim

The visualization program Gsim is a distributed system created by Jensen et al. [2006] capable of visualizing 2D graph simulations from Simulink in a 3D environment. The program is able to use multiple hosts, where one host can run the simulation and another host run the 3D visualization program.

The program includes a Simulink block *3D Visualization Sink* (see Figure A.1), which must be defined with specific information of i.a. what 3D object is to be visualized and the IP of the target host running the 3D visualization program. This Simulink block can handle objects with six degrees of freedom, e.g. the UAV in this thesis, which result in six inputs needed for the visualization program:

- The position of the UAV given in the EF ${}^{e}\Xi^T = [\ {}^{e}x \quad {}^{e}y \quad {}^{e}z\ ]^T$.
- The attitude of the UAV given in the EF ${}^{e}\Theta^T = [\ {}^{e}\phi \quad {}^{e}\theta \quad {}^{e}\psi\ ]^T$.

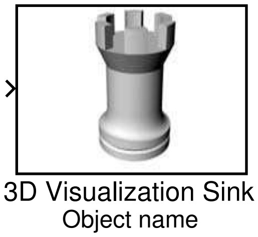
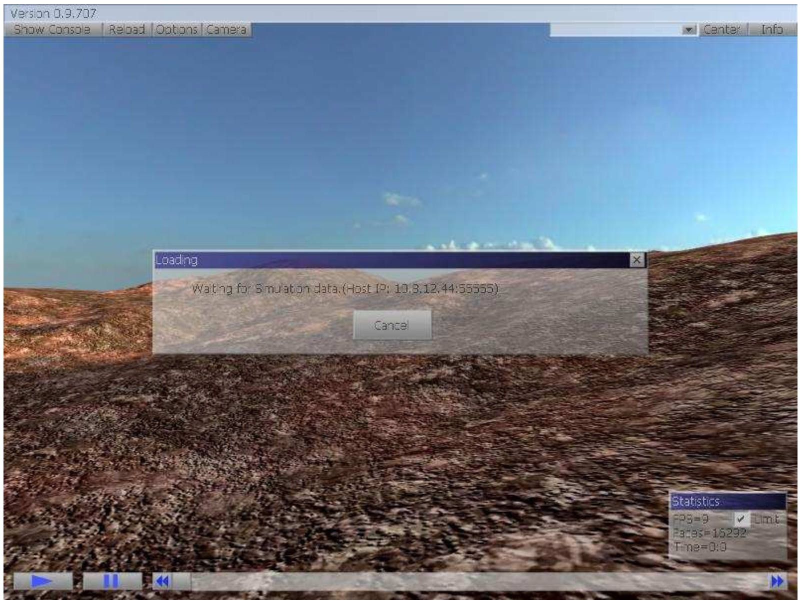


Figure A.1: *Illustration of the* Simulink *block 3D Visualization Sink.*

The 3D Visualization Sink block sends the information received from the actual simulation to the 3D visualization program, which renders the information into 3D. In Figure A.2 the 3D visualization program is shown with no data received from the 3D Visualization Sink block in Simulink. Furthermore, it can be seen that the program shows the host IP, which the 3D Visualization Sink block must target.

In Figure A.3 the 3D visualization program is shown with data received from Simulink, where it can be seen that the program has rendered a 3D helicopter model, which is being moved around and rotated with respect to the data received from the actual simulation. In addition, it

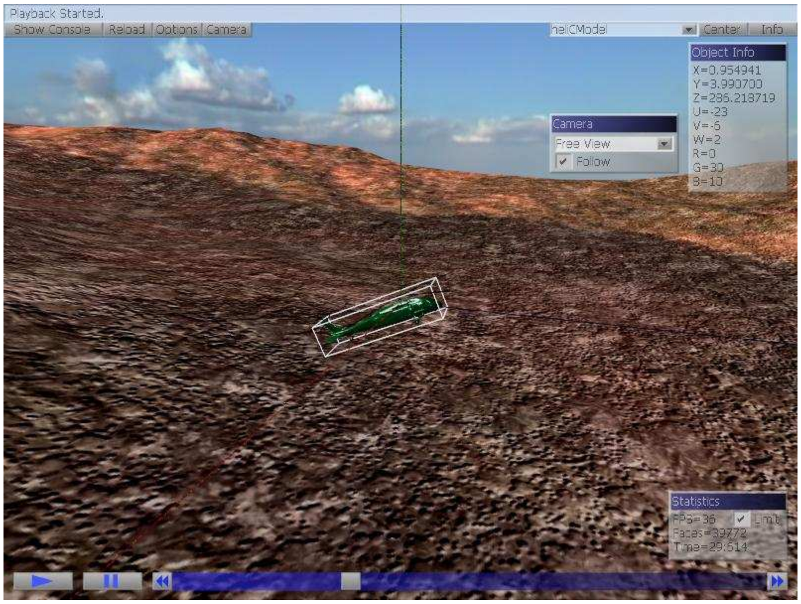


Version 0.9.707

Waiting for Simulation data.(Host IP: 10.3.12.44:55555)

FPS=9 ☑ Limit
Faces=16292
Time=0:0

FIGURE A.2: *Screenshot of the 3D visualization program with no simulation data received. The program shows the IP of the program host that must be targeted by the 3D Visualization Sink block in* SIMULINK.



Playback Started.

heliCModel

X=0.954941
Y=3.990700
Z=286.218719
U=-23
V=-6
W=2
R=0
G=30
B=10

Free View
☑ Follow

FPS=36 ☑ Limit
Faces=39772
Time=29:614

FIGURE A.3: *Screenshot of the 3D visualization program with simulation data received regarding the position and attitude of the UAV. In the top right corner tabs are shown; one for object info and one for camera control.*

can be seen that the 3D visualization program is capable of showing the data regarding the position and attitude of the model helicopter (can be seen in the top right corner in the object info tab). Another feature in the 3D visualization program is the *free view* seen in the camera tab next to the object info tab in the top right corner. The free view feature makes it possible to see the 3D model from any angle and therefore makes it easier to analyze the performance of the simulated system.

A.2 Mapping data from Simulink to Gsim

To be able to use GSIM properly a mapping of the simulated data regarding the position and attitude is necessary. The coordinate system represented in the 3D visualization program has the orientation as illustrated in Figure A.4(b), where as the orientation of the coordinate system of the UAV model is as illustrated in Figure A.4(a).

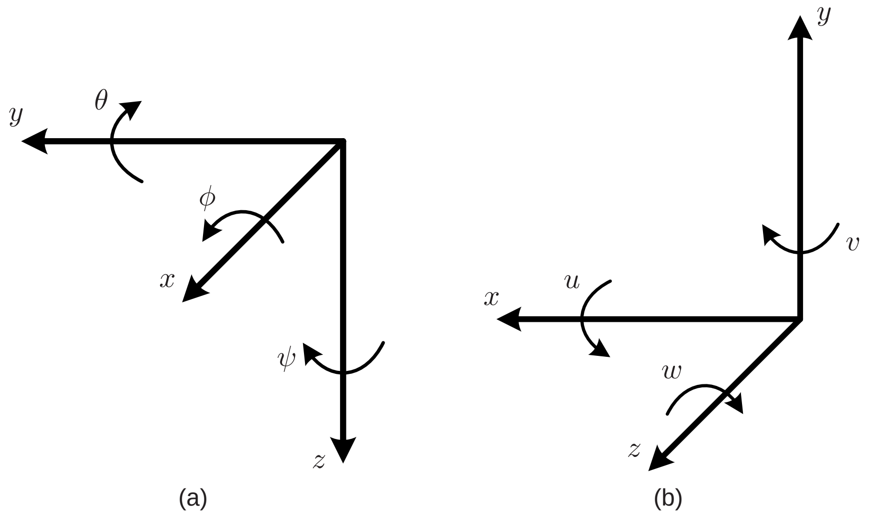


FIGURE A.4: *(a) shows the orientation of the coordinate system in the working model of the UAV. (b) shows the orientation of the coordinate system in the 3D visualization program. Because they are different a mapping from (a) to (b) is necessary.*

Knowing the orientation of the two coordinate systems along with the positive rotations it is rather easy to map the data from the simulation to the 3D visualization program. The mapping is performed in the following manner (described by equations):

$$z_{\text{gsim}} = x_{\text{model}} \tag{A.1}$$

$$x_{\text{gsim}} = y_{\text{model}} \tag{A.2}$$

$$y_{\text{gsim}} = -z_{\text{model}} \tag{A.3}$$

$$w = -\phi\frac{180^\circ}{\pi} \tag{A.4}$$

$$u = -\theta\frac{180^\circ}{\pi} \tag{A.5}$$

$$v = \psi\frac{180^\circ}{\pi} \tag{A.6}$$

Note that the angles mapped are converted from radians to degrees since the 3D visualization program define angles in degrees.

A.3 Rotation problem

There are a known problem with the GSIM program, which compromise the versatile use of the program. The problem reveals it self by for example performing the following rotations of the 3D object:

1. Rotate 90° about the y-axis.
2. Rotate 30° about the x-axis.
3. Rotate 30° about the z-axis.

The result of this rotation of the 3D object is visually seen as a standard 90° rotation about the y-axis, which means that the rotations about the x and z axis have been subtracted and thus resulting in a rotation of 0° about both axes. It can then be concluded that a bug in the program cause the x and z axes to have opposite orientations when a 90° positive rotation about the y-axis is performed. Therefore, to make full use of the visualization program GSIM this bug must be fixed.

Appendix B
Real states versus state estimations

This appendix gives a view of the signals that are to be used as feedback in the control loops. The graphs shown in here are all based on the data collected from the system when initialized in hover.

In Figure B.1 the state estimates for the helicopter position (${}^{e}x$, ${}^{e}y$ and ${}^{e}z$) are shown. As it can be seen from the figure the state estimates have a rough 0.1 m error in general. In addition, the state estimate of ${}^{e}x$ has an initial offset, which ultimately can cause a disturbance in the control loop until the estimator settles.

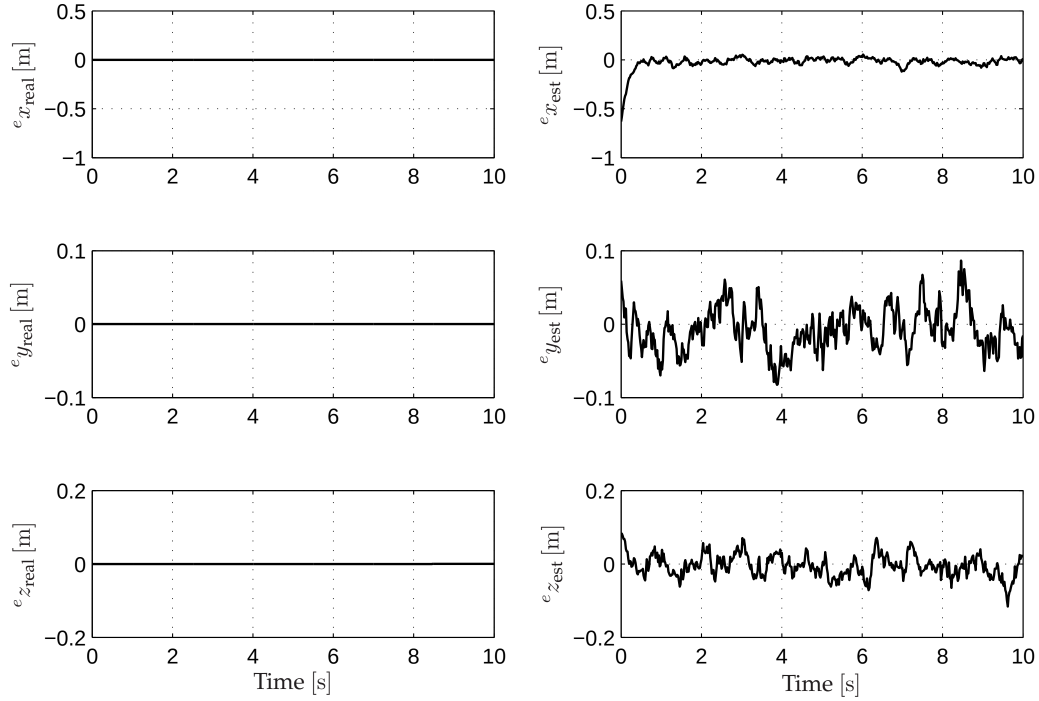


Figure B.1: *Comparison of the real (to the left) and estimated (to the right) position of the helicopter. The shown graphs are obtained by initializing the helicopter in hover.*

In Figure B.2 the state estimates for the attitude (${}^{e}\phi$, ${}^{e}\theta$ and ${}^{e}\psi$) of the helicopter are shown. As it can be seen the estimate of the different angles are rather good except for the yaw angle, which has a rough estimation error of 0.1 rad. Furthermore, it can be seen from the graphs that the alterations seen in the yaw angle estimate is also seen in the roll angle estimation although not as powerful changes.

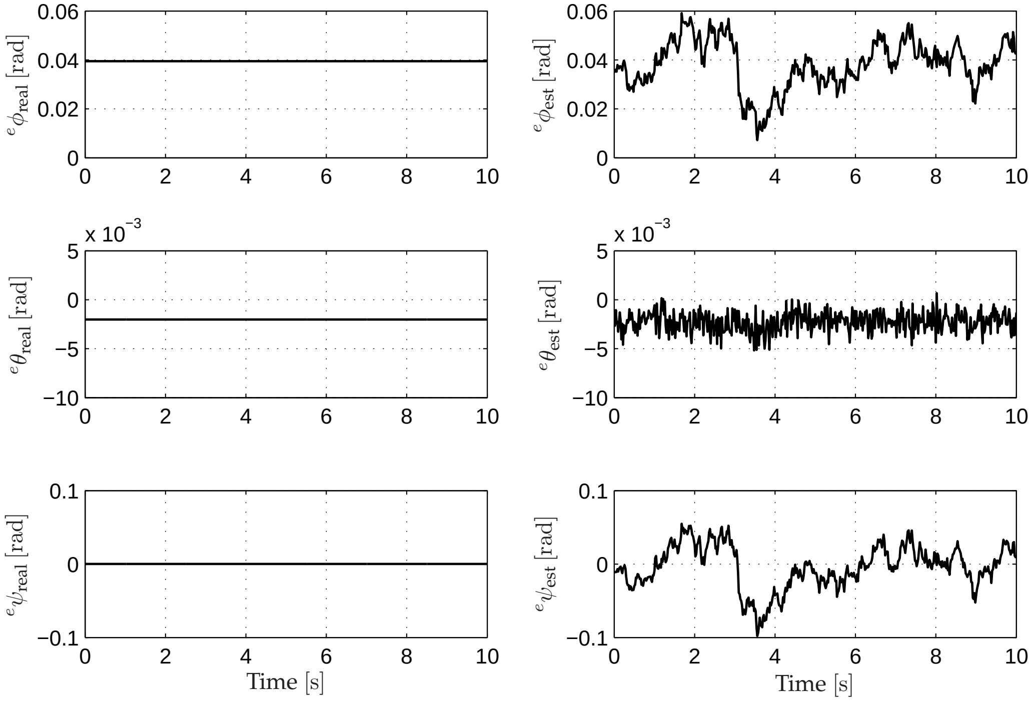


FIGURE B.2: *Comparison of the real (to the left) and estimated (to the right) attitude of the helicopter. The data is obtained by initializing the helicopter in hover.*

In Figure B.3 the state estimates for the helicopters translateral velocities ($^b\dot{x}$, $^b\dot{y}$ and $^b\dot{z}$) are depicted. As it can be seen from the graphs the estimated translateral velocities have a rough estimation error of $0.3\ \frac{\text{m}}{\text{s}}$, which indicate that a compromise may have to be made regarding the definition of hover.

In Figure B.4 the state estimates for the helicopters rotational velocities ($^b\dot{\phi}$, $^b\dot{\theta}$ and $^b\dot{\psi}$) are shown. From the graphs it can be seen that $^b\dot{\phi}$ and $^b\dot{\theta}$ have a rough estimation error of $0.3\ \frac{\text{rad}}{\text{s}}$. In addition, it can be seen that $^b\dot{\psi}$ has a larger estimation error of about $0.8\ \frac{\text{rad}}{\text{s}}$, which may cause the relative large estimation error of the yaw angle $^e\psi$.

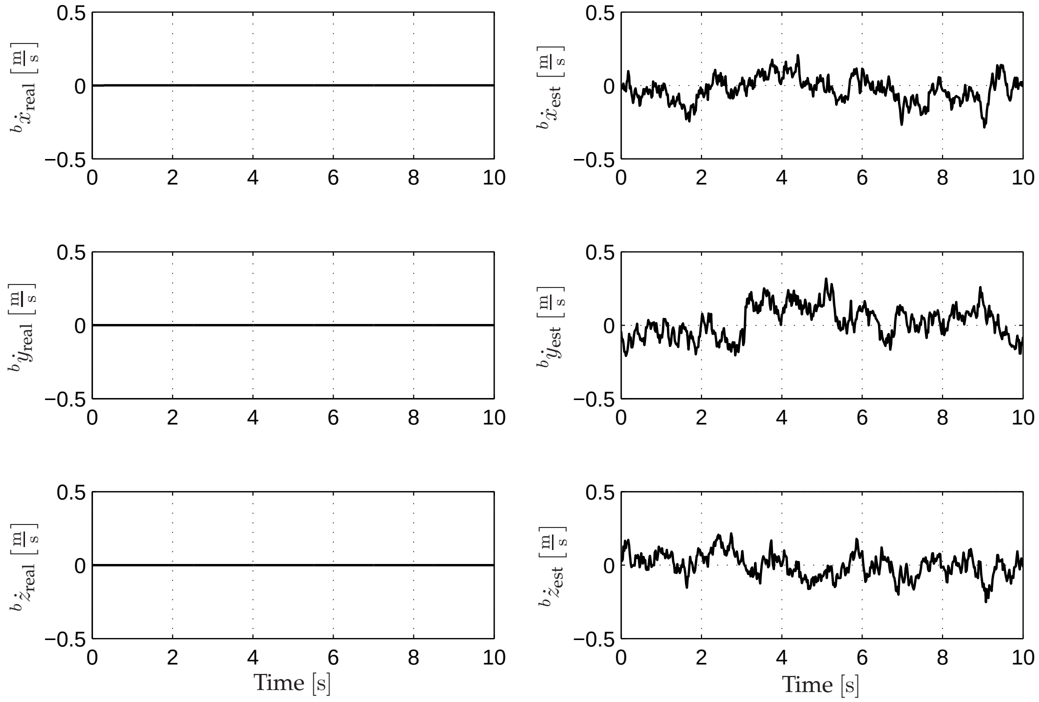


FIGURE B.3: *Comparison of the real (to the left) and the estimated (to the right) translateral velocities of the helicopter. The shown estimates are obtained by initializing the helicopter in hover.*

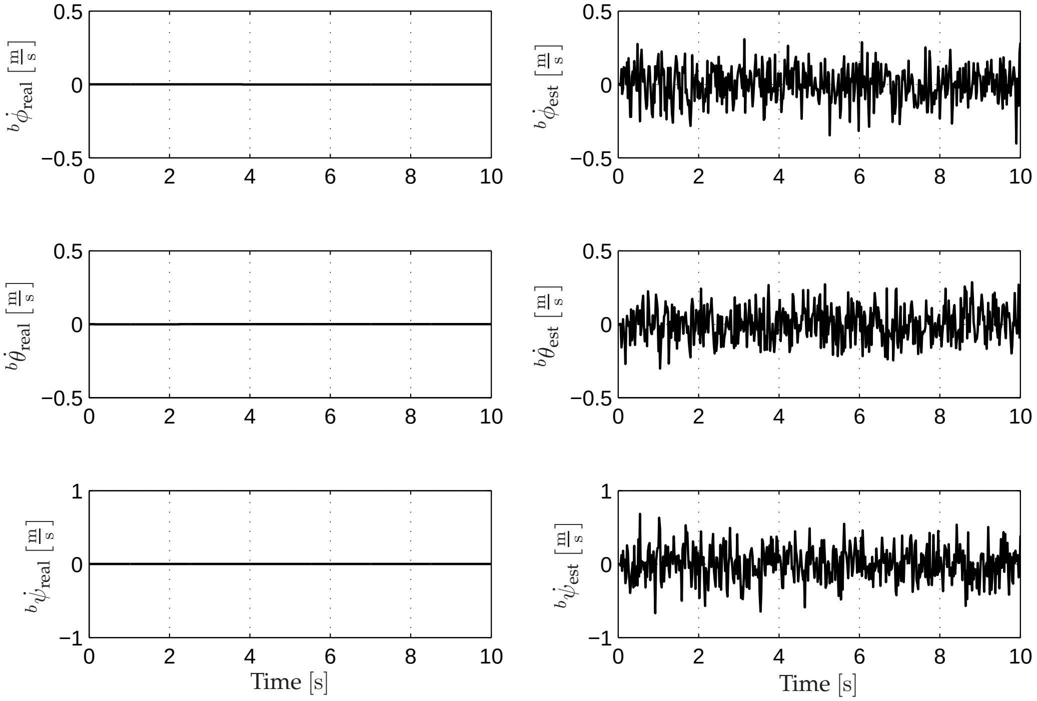


FIGURE B.4: *Comparison of the real (to the left) and the estimated (to the right) rotational velocities of the helicopter. The shown graphs are obtained by initializing the helicopter in hover.*

Appendix C
Linearization of the nonlinear model

The model of the helicopter, which was derived and verified in Hald et al. [2006], is nonlinear but as stated in the introduction to this thesis, there will only be used linear controllers for the helicopter. Therefore the model needs to be linearized in different operation points where the controllers will be optimized for. The linearization is used for both controller types; classic and LQR. The linearization will be performed using a MATLAB© *function developed by Bisgaard [2007] and the linearization method will not be described beyond inputs for the function, which are described in the following section.*

C.1 Linearization conditions

The function, which is called `trim_final_c.m`, takes five inputs:

V**:** The velocity of the helicopter defined initially in the x direction in $\frac{\mathrm{m}}{\mathrm{s}}$.

fp**:** The angle between the x-axis and the vertical offset (positive up) in rad.

bw**:** The angle between the x-axis and the horizontal offset (postive right) in rad.

yaw_ref**:** The yaw angle velocity about the z-axis in $\frac{\mathrm{rad}}{\mathrm{s}}$.

$full$**:** A flag which decides if flapping will be included in the model or rigid body states only.

The above inputs to the linearization function are then used to calculate the velocity helicopter along the three axes as

$$V_x = V\cos(fp)\cos(bw) \tag{C.1}$$

$$V_y = V\cos(fp)\sin(bw) \tag{C.2}$$

$$V_z = -V\sin(bw). \tag{C.3}$$

The output of the function are the linearized model in the form state matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ and the initial values there are needed for developing the controllers for the helicopter.

C.2 State matrices and initial values

Linearized hover values `[`A_0`,`B_0`]=trim_final_c(0,0,0,0,0)`

$$\boldsymbol{A_0} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1.00 & -7.98\cdot10^{-5} & -2.02\cdot10^{-3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.99\cdot10^{-1} & -3.95\cdot10^{-2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2.02\cdot10^{-3} & 3.95\cdot10^{-2} & 9.99\cdot10^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.00 & -7.98\cdot10^{-5} & -2.02\cdot10^{-3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.99\cdot10^{-1} & -3.95\cdot10^{-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.95\cdot10^{-2} & 9.99\cdot10^{-1} \\ 0 & 0 & 0 & 0 & -9.82 & 0 & -2.15\cdot10^{-2} & 1.51\cdot10^{-3} & -1.52\cdot10^{-3} & 1.46\cdot10^{-1} & 1.10 & 0 \\ 0 & 0 & 0 & 9.81 & 7.83\cdot10^{-4} & 0 & -1.42\cdot10^{-3} & -6.83\cdot10^{-2} & 1.15\cdot10^{-2} & -1.10 & 1.46\cdot10^{-1} & 2.78\cdot10^{-1} \\ 0 & 0 & 0 & -3.88\cdot10^{-1} & 0.0198 & 0 & -3.54\cdot10^{-3} & 2.25\cdot10^{-2} & -9.00\cdot10^{-1} & 6.30\cdot10^{-3} & -1.80\cdot10^{-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2.32\cdot10^{-2} & -3.56\cdot10^{-1} & 9.34\cdot10^{-2} & -13.9 & 1.57 & 8.34\cdot10^{-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.14\cdot10^{-1} & -1.44\cdot10^{-2} & 9.30\cdot10^{-3} & -7.87\cdot10^{-1} & -7.00 & 5.47\cdot10^{-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & -4.18\cdot10^{-4} & 7.42\cdot10^{-1} & -8.13\cdot10^{-2} & -2.91\cdot10^{-1} & -5.12\cdot10^{-2} & -4.43 \end{bmatrix} \tag{C.4}$$

$$\boldsymbol{B}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1.93\cdot10^{-1} & -4.45\cdot10^{-2} & 7.03 & 0 \\ 1.33 & -7.03 & -4.45\cdot10^{-2} & -2.27\cdot10^{-1} \\ -9.85\cdot10^{1} & 0 & 0 & 0 \\ 1.08\cdot10^{1} & -8.51\cdot10^{1} & -8.17\cdot10^{-1} & -6.97\cdot10^{-1} \\ 5.65\cdot10^{-1} & 4.08\cdot10^{-1} & -4.25\cdot10^{1} & -5.20\cdot10^{-2} \\ -7.54\cdot10^{1} & 0 & 0 & 3.62 \end{bmatrix} \quad \begin{bmatrix} {}^{b}\dot{x} \\ {}^{b}\dot{y} \\ {}^{b}\dot{z} \\ {}^{b}\dot{\phi} \\ {}^{b}\dot{\theta} \\ {}^{b}\dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} {}^{e}x \\ {}^{e}y \\ {}^{e}z \\ {}^{e}\phi \\ {}^{e}\theta \\ {}^{e}\psi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3.95\cdot10^{-2} \\ -2.02\cdot10^{-3} \\ 1.40\cdot10^{-1} \end{bmatrix} \tag{C.5}$$

Linearized $10\,\frac{\mathrm{m}}{\mathrm{s}}$ forward flight values `[`A_1`,`B_1`]=trim_final_c(10,0,0,0,0)`

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1.00 & -7.36\cdot10^{-4} & -2.75\cdot10^{-2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.75\cdot10^{-1} & 0 & 1.00\cdot10^{1} & 0 & 1.00 & -2.68\cdot10^{-2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.00\cdot10^{1} & 0 & 2.75\cdot10^{-2} & 2.68\cdot10^{-2} & 1.00 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.00 & -7.36\cdot10^{-4} & -2.75\cdot10^{-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.00 & -2.68\cdot10^{-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.68\cdot10^{-2} & 1.00 \\ 0 & 0 & 0 & 0 & -9.82 & 0 & -6.34\cdot10^{-2} & 3.01\cdot10^{-4} & 1.08\cdot10^{-2} & 1.31\cdot10^{-1} & 1.43 & -7.36\cdot10^{-3} \\ 0 & 0 & 0 & 9.81 & 7.23\cdot10^{-3} & 0 & -1.44\cdot10^{-2} & -9.81\cdot10^{-2} & 2.23\cdot10^{-2} & -1.43 & 1.13\cdot10^{-1} & -9.68 \\ 0 & 0 & 0 & -2.63\cdot10^{-1} & 2.70\cdot10^{-1} & 0 & -3.24\cdot10^{-1} & 1.89\cdot10^{-2} & -1.84 & 1.58\cdot10^{-1} & 1.19\cdot10^{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3.24\cdot10^{-2} & -4.09\cdot10^{-1} & 1.78\cdot10^{-1} & -1.42\cdot10^{1} & 1.33 & 9.31\cdot10^{-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 8.11\cdot10^{-2} & -6.82\cdot10^{-3} & -4.06\cdot10^{-2} & -7.29\cdot10^{-1} & -7.20 & 2.40\cdot10^{-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 3.43\cdot10^{-1} & 1.29 & 2.74\cdot10^{-1} & -1.44\cdot10^{-1} & -4.51\cdot10^{-1} & -5.08 \end{bmatrix} \tag{C.6}$$

$$\boldsymbol{B}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1.25 & -7.78\cdot10^{-2} & 6.9 & 0 \\ 7.74\cdot10^{-1} & -6.93 & -2.03\cdot10^{-1} & -2.29\cdot10^{-1} \\ -1.08\cdot10^{2} & 9.65\cdot10^{-3} & 1.31\cdot10^{1} & 0 \\ 4.26 & -8.43\cdot10^{1} & -1.88 & -6.79\cdot10^{-1} \\ 8.02 & 5.44\cdot10^{-1} & -4.24\cdot10^{1} & -2.41\cdot10^{-2} \\ -4.70\cdot10^{1} & 8.82\cdot10^{-2} & -2.14 & 3.65 \end{bmatrix} \quad \begin{bmatrix} {}^{b}\dot{x} \\ {}^{b}\dot{y} \\ {}^{b}\dot{z} \\ {}^{b}\dot{\phi} \\ {}^{b}\dot{\theta} \\ {}^{b}\dot{\psi} \end{bmatrix} = \begin{bmatrix} 1.00 \\ -7.36\cdot10^{-3} \\ -2.75\cdot10^{-1} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} {}^{e}x \\ {}^{e}y \\ {}^{e}z \\ {}^{e}\phi \\ {}^{e}\theta \\ {}^{e}\psi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2.61\cdot10^{-2} \\ -2.75\cdot10^{-2} \\ 0 \end{bmatrix} \tag{C.7}$$

Appendix D
Calculation of $\tilde{\mathcal{K}}$

The use of the observer based gain scheduling presented by Bendtsen et al. [2005] implies that it is necessary to design $\tilde{\mathcal{K}}$ such that the interconnection $\mathcal{K} \star \tilde{\mathcal{K}}$ is an identity system. This means that the outputs of the system $\mathcal{K} \star \tilde{\mathcal{K}}$ is identical to the inputs, and that the augmented state vectors $\hat{\boldsymbol{\xi}}_0$ and $\tilde{\boldsymbol{\xi}}$ are equal to each other. This appendix deals with the design of $\tilde{\mathcal{K}}$ such that these criteria are fulfilled.

D.1 Calculation method

The method used for the determination of $\tilde{\mathcal{K}}$ is the general theory for calculation of interconnection between two systems, Zhou et al. [1996]. In this case the two systems are $\mathcal{K}$ and $\tilde{\mathcal{K}}$, which can be generally defined as

$$\mathcal{K} = \left[\begin{array}{c|cc} \boldsymbol{A} & \boldsymbol{B}_1 & \boldsymbol{B}_2 \\ \hline \boldsymbol{C}_1 & \boldsymbol{D}_{11} & \boldsymbol{D}_{12} \\ \boldsymbol{C}_2 & \boldsymbol{D}_{21} & \boldsymbol{D}_{22} \end{array}\right] \quad \tilde{\mathcal{K}} = \left[\begin{array}{c|cc} \boldsymbol{A}_K & \boldsymbol{B}_{K1} & \boldsymbol{B}_{K2} \\ \hline \boldsymbol{C}_{K1} & \boldsymbol{D}_{K11} & \boldsymbol{D}_{K12} \\ \boldsymbol{C}_{K2} & \boldsymbol{D}_{K21} & \boldsymbol{D}_{K22} \end{array}\right] \quad .$$

The interconnection (also called the *Redheffer Star-product*) is then defined as

$$\mathcal{S}(\mathcal{K},\tilde{\mathcal{K}}) = \left[\begin{array}{c|c} \bar{\boldsymbol{A}} & \bar{\boldsymbol{B}} \\ \hline \bar{\boldsymbol{C}} & \bar{\boldsymbol{D}} \end{array}\right] = \left[\begin{array}{cc|cc} \bar{\boldsymbol{A}}_{11} & \bar{\boldsymbol{A}}_{12} & \bar{\boldsymbol{B}}_{11} & \bar{\boldsymbol{B}}_{12} \\ \bar{\boldsymbol{A}}_{21} & \bar{\boldsymbol{A}}_{22} & \bar{\boldsymbol{B}}_{21} & \bar{\boldsymbol{B}}_{22} \\ \hline \bar{\boldsymbol{C}}_{11} & \bar{\boldsymbol{C}}_{12} & \bar{\boldsymbol{D}}_{11} & \bar{\boldsymbol{D}}_{12} \\ \bar{\boldsymbol{C}}_{21} & \bar{\boldsymbol{C}}_{22} & \bar{\boldsymbol{D}}_{21} & \bar{\boldsymbol{D}}_{22} \end{array}\right] \quad ,$$

where

$$\bar{\boldsymbol{A}} = \begin{bmatrix} \bar{\boldsymbol{A}}_{11} & \bar{\boldsymbol{A}}_{12} \\ \bar{\boldsymbol{A}}_{21} & \bar{\boldsymbol{A}}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} + \boldsymbol{B}_2\tilde{\boldsymbol{R}}^{-1}\boldsymbol{D}_{K11}\boldsymbol{C}_2 & \boldsymbol{B}_2\tilde{\boldsymbol{R}}^{-1}\boldsymbol{C}_{K1} \\ \boldsymbol{B}_{K1}\boldsymbol{R}^{-1}\boldsymbol{C}_2 & \boldsymbol{A}_K + \boldsymbol{B}_{K1}\boldsymbol{R}^{-1}\boldsymbol{D}_{22}\boldsymbol{C}_{K1} \end{bmatrix} \tag{D.1}$$

$$\bar{\boldsymbol{B}} = \begin{bmatrix} \bar{\boldsymbol{B}}_{11} & \bar{\boldsymbol{B}}_{12} \\ \bar{\boldsymbol{B}}_{21} & \bar{\boldsymbol{B}}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_1 + \boldsymbol{B}_2\tilde{\boldsymbol{R}}^{-1}\boldsymbol{D}_{K11}\boldsymbol{D}_{21} & \boldsymbol{B}_2\tilde{\boldsymbol{R}}^{-1}\boldsymbol{D}_{K12} \\ \boldsymbol{B}_{K1}\boldsymbol{R}^{-1}\boldsymbol{D}_{21} & \boldsymbol{B}_{K2} + \boldsymbol{B}_{K1}\boldsymbol{R}^{-1}\boldsymbol{D}_{22}\boldsymbol{D}_{K12} \end{bmatrix} \tag{D.2}$$

$$\bar{\boldsymbol{C}} = \begin{bmatrix} \bar{\boldsymbol{C}}_{11} & \bar{\boldsymbol{C}}_{12} \\ \bar{\boldsymbol{C}}_{21} & \bar{\boldsymbol{C}}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}_1 + \boldsymbol{D}_{12}\boldsymbol{D}_{K11}\boldsymbol{R}^{-1}\boldsymbol{C}_2 & \boldsymbol{D}_{12}\tilde{\boldsymbol{R}}^{-1}\boldsymbol{C}_{K1} \\ \boldsymbol{D}_{K21}\boldsymbol{R}^{-1}\boldsymbol{C}_2 & \boldsymbol{C}_{K2} + \boldsymbol{D}_{K21}\boldsymbol{R}^{-1}\boldsymbol{D}_{22}\boldsymbol{C}_{K1} \end{bmatrix} \tag{D.3}$$

$$\bar{\boldsymbol{D}} = \begin{bmatrix} \bar{\boldsymbol{D}}_{11} & \bar{\boldsymbol{D}}_{12} \\ \bar{\boldsymbol{D}}_{21} & \bar{\boldsymbol{D}}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}_{11} + \boldsymbol{D}_{12}\boldsymbol{D}_{K11}\boldsymbol{R}^{-1}\boldsymbol{D}_{21} & \boldsymbol{D}_{21}\tilde{\boldsymbol{R}}^{-1}\boldsymbol{D}_{K12} \\ \boldsymbol{D}_{K21}\boldsymbol{R}^{-1}\boldsymbol{D}_{21} & \boldsymbol{D}_{K22} + \boldsymbol{D}_{K21}\boldsymbol{R}^{-1}\boldsymbol{D}_{22}\boldsymbol{D}_{K12} \end{bmatrix} \tag{D.4}$$

$$\boldsymbol{R} = \boldsymbol{I} - \boldsymbol{D}_{22}\boldsymbol{D}_{K11}, \quad \tilde{\boldsymbol{R}} = \boldsymbol{I} - \boldsymbol{D}_{K11}\boldsymbol{D}_{22} \quad . \tag{D.5}$$

Since the structure of $\tilde{\mathcal{K}}$ is not known it is necessary to determine it, based on the criteria defined for the interconnection, which will be done in the following section.

D.2 Design of $\tilde{\mathcal{K}}$

As the first step in designing $\tilde{\mathcal{K}}$ it is necessary to identify the structure of $\mathcal{K}$, which is found to be

$$\mathcal{K} = \left[\begin{array}{cc|cc} \boldsymbol{A}_0 - \boldsymbol{B}_0\boldsymbol{L}_0 - \boldsymbol{F}_0\boldsymbol{C}_y & \boldsymbol{B}_0 & \boldsymbol{F}_0 & \boldsymbol{B}_0 \\ \mathbf{0} & \mathbf{0} & \boldsymbol{L}_{I0} & r\boldsymbol{I} \\ \hline -\boldsymbol{L}_0 & \boldsymbol{I} & \mathbf{0} & \boldsymbol{I} \\ -\boldsymbol{C}_y & \mathbf{0} & \boldsymbol{I} & \mathbf{0} \end{array}\right] = \left[\begin{array}{c|cc} \boldsymbol{A} & \boldsymbol{B}_1 & \boldsymbol{B}_2 \\ \hline \boldsymbol{C}_1 & \boldsymbol{D}_{11} & \boldsymbol{D}_{12} \\ \boldsymbol{C}_2 & \boldsymbol{D}_{21} & \boldsymbol{D}_{22} \end{array}\right] . \tag{D.6}$$

Having determined $\mathcal{K}$ it is possible to start designing $\tilde{\mathcal{K}}$, which is done in a backwards manner.

D.2.1 Determination of $\boldsymbol{D}_{K11}$, $\boldsymbol{D}_{K12}$, $\boldsymbol{D}_{K21}$ and $\boldsymbol{D}_{K22}$

From the criterion defining, that it should be possible to perform direct feed through from input to output, makes it possible to define $\bar{\boldsymbol{D}}$ as

$$\bar{\boldsymbol{D}} = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \boldsymbol{I} & \mathbf{0} \end{bmatrix} . \tag{D.7}$$

This results in solving the equations given in (D.4) with respect to $\boldsymbol{D}_{K11}$, $\boldsymbol{D}_{K12}$, $\boldsymbol{D}_{K21}$ and $\boldsymbol{D}_{K22}$. But first the values of $\boldsymbol{R}$ and $\tilde{\boldsymbol{R}}$ are identified to ease the calculations:

$$\boldsymbol{R} = \boldsymbol{I} - \boldsymbol{D}_{22}\boldsymbol{D}_{K11} = \boldsymbol{I} - \mathbf{0}\boldsymbol{D}_{K11} = \boldsymbol{I} \tag{D.8}$$

$$\tilde{\boldsymbol{R}} = \boldsymbol{I} - \boldsymbol{D}_{K11}\boldsymbol{D}_{22} = \boldsymbol{I} - \boldsymbol{D}_{K11}\mathbf{0} = \boldsymbol{I} \ . \tag{D.9}$$

With $\boldsymbol{R}$ and $\tilde{\boldsymbol{R}}$ determined to be identity matrices these can be overlooked when solving the equations.

By solving the equations for $\bar{\boldsymbol{D}}_{11}$, $\bar{\boldsymbol{D}}_{12}$, $\bar{\boldsymbol{D}}_{21}$ and $\bar{\boldsymbol{D}}_{22}$ the following is obtained

$$\bar{\boldsymbol{D}}_{11} = \mathbf{0} = \boldsymbol{D}_{11} + \boldsymbol{D}_{12}\boldsymbol{D}_{K11}\boldsymbol{D}_{21} = \mathbf{0} + \boldsymbol{I}\boldsymbol{D}_{K11}\boldsymbol{I} \Rightarrow \boldsymbol{D}_{K11} = \mathbf{0} \tag{D.10}$$

$$\bar{\boldsymbol{D}}_{12} = \boldsymbol{I} = \boldsymbol{D}_{21}\boldsymbol{D}_{K12} = \boldsymbol{I}\boldsymbol{D}_{K12} \Rightarrow \boldsymbol{D}_{K12} = \boldsymbol{I} \tag{D.11}$$

$$\bar{\boldsymbol{D}}_{21} = \boldsymbol{I} = \boldsymbol{D}_{K21}\boldsymbol{D}_{21} = \boldsymbol{D}_{K21}\boldsymbol{I} \Rightarrow \boldsymbol{D}_{K21} = \boldsymbol{I} \tag{D.12}$$

$$\bar{\boldsymbol{D}}_{22} = \mathbf{0} = \boldsymbol{D}_{K22} + \boldsymbol{D}_{K21}\boldsymbol{D}_{22}\boldsymbol{D}_{K12} = \boldsymbol{D}_{K22} + \boldsymbol{I}\mathbf{0}\boldsymbol{I} \Rightarrow \boldsymbol{D}_{K22} = \mathbf{0} \ . \tag{D.13}$$

With the above solutions $\tilde{\mathcal{K}}$ can be updated with respect to $\boldsymbol{D}_{K11}$, $\boldsymbol{D}_{K12}$, $\boldsymbol{D}_{K21}$ and $\boldsymbol{D}_{K22}$:

$$\tilde{\mathcal{K}} = \left[\begin{array}{c|cc} \boldsymbol{A}_K & \boldsymbol{B}_{K1} & \boldsymbol{B}_{K2} \\ \hline \boldsymbol{C}_{K1} & \mathbf{0} & \boldsymbol{I} \\ \boldsymbol{C}_{K2} & \boldsymbol{I} & \mathbf{0} \end{array}\right] . \tag{D.14}$$

D.2.2 Determination of $\boldsymbol{C}_{K1}$ and $\boldsymbol{C}_{K2}$

Based on the criterion that the output must be equal to the input when $\hat{\boldsymbol{\xi}}_0 = \tilde{\boldsymbol{\xi}}$ results in the new criterion that $\bar{\boldsymbol{C}}_{11} + \bar{\boldsymbol{C}}_{12} = \mathbf{0}$, and $\bar{\boldsymbol{C}}_{21} + \bar{\boldsymbol{C}}_{22} = \mathbf{0}$. This leads to solving the equations given in

(D.3) with respect to $\boldsymbol{C}_{K1}$ and $\boldsymbol{C}_{K2}$, and the aforementioned criteria:

$$\begin{aligned}
\bar{\boldsymbol{C}}_{11} &= \boldsymbol{C}_1 + \boldsymbol{D}_{12}\boldsymbol{D}_{K11}\boldsymbol{C}_2 = \begin{bmatrix} -\boldsymbol{L}_0 & \boldsymbol{I} \end{bmatrix} + \boldsymbol{I}\boldsymbol{0}\begin{bmatrix} -\boldsymbol{C}_y & \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{L}_0 & \boldsymbol{I} \end{bmatrix} \\
\bar{\boldsymbol{C}}_{12} &= \boldsymbol{D}_{12}\boldsymbol{C}_{K1} = \boldsymbol{I}\boldsymbol{C}_{K1} = \boldsymbol{C}_{K1} \\
&\Downarrow
\end{aligned}$$

$$\bar{\boldsymbol{C}}_{11} + \bar{\boldsymbol{C}}_{12} = \boldsymbol{0} = \begin{bmatrix} -\boldsymbol{L}_0 & \boldsymbol{I} \end{bmatrix} + \boldsymbol{C}_{K1} \Rightarrow \boldsymbol{C}_{K1} = \begin{bmatrix} \boldsymbol{L}_0 & -\boldsymbol{I} \end{bmatrix} \quad . \tag{D.15}$$

The same procedure is used for determine $\boldsymbol{C}_{K2}$:

$$\begin{aligned}
\bar{\boldsymbol{C}}_{21} &= \boldsymbol{D}_{K21}\boldsymbol{C}_2 = \boldsymbol{I}\begin{bmatrix} -\boldsymbol{C}_y & \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{C}_y & \boldsymbol{0} \end{bmatrix} \\
\bar{\boldsymbol{C}}_{22} &= \boldsymbol{C}_{K2} + \boldsymbol{D}_{K21}\boldsymbol{D}_{22}\boldsymbol{C}_{K1} = \boldsymbol{C}_{K2} + \boldsymbol{I}\boldsymbol{0}\begin{bmatrix} \boldsymbol{L}_0 & -\boldsymbol{I} \end{bmatrix} = \boldsymbol{C}_{K2} \\
&\Downarrow
\end{aligned}$$

$$\bar{\boldsymbol{C}}_{21} + \bar{\boldsymbol{C}}_{22} = \boldsymbol{0} = \begin{bmatrix} -\boldsymbol{C}_y & \boldsymbol{0} \end{bmatrix} + \boldsymbol{C}_{K2} \Rightarrow \boldsymbol{C}_{K2} = \begin{bmatrix} \boldsymbol{C}_y & \boldsymbol{0} \end{bmatrix} \quad . \tag{D.16}$$

The updated version of $\tilde{\mathcal{K}}$ is then

$$\tilde{\mathcal{K}} = \left[\begin{array}{cc|cc} \multicolumn{2}{c|}{\boldsymbol{A}_K} & \boldsymbol{B}_{K1} & \boldsymbol{B}_{K2} \\ \hline \boldsymbol{L}_0 & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{C}_y & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \end{array}\right] \quad . \tag{D.17}$$

D.2.3 Determination of $\boldsymbol{B}_{K1}$ and $\boldsymbol{B}_{K2}$

To make sure that the augmented state vectors $\hat{\boldsymbol{\xi}}_0$ and $\tilde{\boldsymbol{\xi}}$ are equally affected by the interconnected systems inputs it can be defined that $\bar{\boldsymbol{B}}_{11} = \bar{\boldsymbol{B}}_{21}$ and $\bar{\boldsymbol{B}}_{12} = \bar{\boldsymbol{B}}_{22}$, which leads to solving the equations given in (D.2):

$$\begin{aligned}
\bar{\boldsymbol{B}}_{11} &= \boldsymbol{B}_1 + \boldsymbol{B}_2\boldsymbol{D}_{K11}\boldsymbol{D}_{21} = \begin{bmatrix} \boldsymbol{F}_0 \\ \boldsymbol{L}_{I0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_0 \\ r\boldsymbol{I} \end{bmatrix}\boldsymbol{0}\boldsymbol{I} = \begin{bmatrix} \boldsymbol{F}_0 \\ \boldsymbol{L}_{I0} \end{bmatrix} \\
\bar{\boldsymbol{B}}_{12} &= \boldsymbol{B}_2\boldsymbol{D}_{K12} = \begin{bmatrix} \boldsymbol{B}_0 \\ r\boldsymbol{I} \end{bmatrix}\boldsymbol{I} = \begin{bmatrix} \boldsymbol{B}_0 \\ r\boldsymbol{I} \end{bmatrix}
\end{aligned}$$

Having determined $\bar{\boldsymbol{B}}_{11}$ and $\bar{\boldsymbol{B}}_{12}$ it is relatively easy to determine $\bar{\boldsymbol{B}}_{21}$ and $\bar{\boldsymbol{B}}_{22}$ as

$$\bar{\boldsymbol{B}}_{21} = \boldsymbol{B}_{K1}\boldsymbol{D}_{21} = \boldsymbol{B}_{K1}\boldsymbol{I} = \boldsymbol{B}_{K1} = \begin{bmatrix} \boldsymbol{F}_0 \\ \boldsymbol{L}_{I0} \end{bmatrix} \tag{D.18}$$

$$\bar{\boldsymbol{B}}_{22} = \boldsymbol{B}_{K2} + \boldsymbol{B}_{K1}\boldsymbol{D}_{22}\boldsymbol{D}_{K12} = \boldsymbol{B}_{K2} + \begin{bmatrix} \boldsymbol{F}_0 \\ \boldsymbol{L}_{I0} \end{bmatrix}\boldsymbol{0}\boldsymbol{0} = \boldsymbol{B}_{K2} = \begin{bmatrix} \boldsymbol{B}_0 \\ r\boldsymbol{I} \end{bmatrix} \quad . \tag{D.19}$$

With the determination of $\boldsymbol{B}_{K1}$ and $\boldsymbol{B}_{K2}$, $\tilde{\mathcal{K}}$ is now

$$\tilde{\mathcal{K}} = \left[\begin{array}{cc|cc} \multicolumn{2}{c|}{} & \boldsymbol{F}_0 & \boldsymbol{B}_0 \\ \multicolumn{2}{c|}{\boldsymbol{A}_K} & \boldsymbol{L}_{I0} & r\boldsymbol{I} \\ \hline \boldsymbol{L}_0 & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{C}_y & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \end{array}\right] \quad . \tag{D.20}$$

D.2.4 Determination of $\boldsymbol{A}_K$

The criterion for the determination of $\boldsymbol{A}_K$ is that when $\hat{\boldsymbol{\xi}}_0 = \tilde{\boldsymbol{\xi}}$ they will stay that way regardless of the inputs to the system. This criterion can be rephrased to

$$\begin{aligned} \bar{\boldsymbol{A}}_{11}\hat{\boldsymbol{\xi}}_0 + \bar{\boldsymbol{A}}_{12}\tilde{\boldsymbol{\xi}} &= \bar{\boldsymbol{A}}_{21}\hat{\boldsymbol{\xi}}_0 + \bar{\boldsymbol{A}}_{22}\tilde{\boldsymbol{\xi}} && \text{(D.21)} \\ &\Downarrow \\ \bar{\boldsymbol{A}}_{11} + \bar{\boldsymbol{A}}_{12} &= \bar{\boldsymbol{A}}_{21} + \bar{\boldsymbol{A}}_{22} \ , \ \text{ for } \ \hat{\boldsymbol{\xi}}_0 = \tilde{\boldsymbol{\xi}} \ . && \text{(D.22)} \end{aligned}$$

With the above in mind the equations given in (D.1) can be solved with respect to determining $\boldsymbol{A}_K$:

$$\begin{aligned} \bar{\boldsymbol{A}}_{11} &= \boldsymbol{A} + \boldsymbol{B}_2\boldsymbol{D}_{K11}\boldsymbol{C}_2 = \begin{bmatrix} \boldsymbol{A}_0 - \boldsymbol{B}_0\boldsymbol{L}_0 - \boldsymbol{F}_0\boldsymbol{C}_y & \boldsymbol{B}_0 \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_0 \\ r\boldsymbol{I} \end{bmatrix} \boldsymbol{0} \ [-\boldsymbol{C}_y \quad \boldsymbol{0}] \\ &= \begin{bmatrix} \boldsymbol{A}_0 - \boldsymbol{B}_0\boldsymbol{L}_0 - \boldsymbol{F}_0\boldsymbol{C}_y & \boldsymbol{B}_0 \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \\ \bar{\boldsymbol{A}}_{12} &= \boldsymbol{B}_2\boldsymbol{C}_{K1} = \begin{bmatrix} \boldsymbol{B}_0 \\ r\boldsymbol{I} \end{bmatrix} \ [\boldsymbol{L}_0 \quad -\boldsymbol{I}] \ = \begin{bmatrix} \boldsymbol{B}_0\boldsymbol{L}_0 & -\boldsymbol{B}_0 \\ r\boldsymbol{L}_0 & -r\boldsymbol{I} \end{bmatrix} \\ \bar{\boldsymbol{A}}_{21} &= \boldsymbol{B}_{K1}\boldsymbol{C}_2 = \begin{bmatrix} \boldsymbol{F}_0 \\ \boldsymbol{L}_{I0} \end{bmatrix} \ [-\boldsymbol{C}_y \quad \boldsymbol{0}] \ = \begin{bmatrix} -\boldsymbol{F}_0\boldsymbol{C}_y & \boldsymbol{0} \\ -\boldsymbol{L}_{I0}\boldsymbol{C}_y & \boldsymbol{0} \end{bmatrix} \\ \bar{\boldsymbol{A}}_{22} &= \boldsymbol{A}_K + \boldsymbol{B}_{K1}\boldsymbol{D}_{22}\boldsymbol{C}_{K1} = \boldsymbol{A}_K + \begin{bmatrix} \boldsymbol{F}_0 \\ \boldsymbol{L}_{I0} \end{bmatrix} \boldsymbol{0} \ [\boldsymbol{L}_0 \quad -\boldsymbol{I}] \ = \boldsymbol{A}_K \ . && \text{(D.23)} \end{aligned}$$

As it can be seen from the above calculations the only unknown is the matrix $\boldsymbol{A}_K$. Therefore, the equality $\bar{\boldsymbol{A}}_{11} + \bar{\boldsymbol{A}}_{12} = \bar{\boldsymbol{A}}_{21} + \bar{\boldsymbol{A}}_{22}$ can be solved with respect to $\boldsymbol{A}_K$:

$$\begin{aligned} \bar{\boldsymbol{A}}_{11} + \bar{\boldsymbol{A}}_{12} &= \begin{bmatrix} \boldsymbol{A}_0 - \boldsymbol{B}_0\boldsymbol{L}_0 - \boldsymbol{F}_0\boldsymbol{C}_y & \boldsymbol{B}_0 \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_0\boldsymbol{L}_0 & -\boldsymbol{B}_0 \\ r\boldsymbol{L}_0 & -r\boldsymbol{I} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{A}_0 - \boldsymbol{F}_0\boldsymbol{C}_y & \boldsymbol{0} \\ r\boldsymbol{L}_0 & -r\boldsymbol{I} \end{bmatrix} && \text{(D.24)} \\ \bar{\boldsymbol{A}}_{21} + \bar{\boldsymbol{A}}_{22} &= \begin{bmatrix} -\boldsymbol{L}_0\boldsymbol{C}_y & \boldsymbol{0} \\ -\boldsymbol{L}_{I0}\boldsymbol{C}_y & \boldsymbol{0} \end{bmatrix} + \boldsymbol{A}_K = \begin{bmatrix} \boldsymbol{A}_0 - \boldsymbol{F}_0\boldsymbol{C}_y & \boldsymbol{0} \\ r\boldsymbol{L}_0 & -r\boldsymbol{I} \end{bmatrix} \\ &\Downarrow \\ \boldsymbol{A}_K &= \begin{bmatrix} \boldsymbol{A}_0 - \boldsymbol{F}_0\boldsymbol{C}_y & \boldsymbol{0} \\ r\boldsymbol{L}_0 & -r\boldsymbol{I} \end{bmatrix} - \begin{bmatrix} -\boldsymbol{F}_0\boldsymbol{C}_y & \boldsymbol{0} \\ -\boldsymbol{L}_{I0}\boldsymbol{C}_y & \boldsymbol{0} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{A}_0 & \boldsymbol{0} \\ r\boldsymbol{L}_0 + \boldsymbol{L}_{I0}\boldsymbol{C}_y & -r\boldsymbol{I} \end{bmatrix} \ . && \text{(D.25)} \end{aligned}$$

Having determined $\boldsymbol{A}_K$ the design of $\tilde{\mathcal{K}}$ is complete:

$$\tilde{\mathcal{K}} = \left[\begin{array}{cc|cc} \boldsymbol{A}_0 & \mathbf{0} & \boldsymbol{F}_0 & \boldsymbol{B}_0 \\ r\boldsymbol{L}_0 + \boldsymbol{L}_{I0}\boldsymbol{C}_y & -rI & \boldsymbol{L}_{I0} & rI \\ \hline \boldsymbol{L}_0 & -\boldsymbol{I} & \mathbf{0} & \boldsymbol{I} \\ \boldsymbol{C}_y & \mathbf{0} & \boldsymbol{I} & \mathbf{0} \end{array}\right] , \tag{D.26}$$

where the interconnection $\mathcal{K} \star \tilde{\mathcal{K}}$ is given as

$$\mathcal{K} \star \tilde{\mathcal{K}} = \left[\begin{array}{cccc|cc} \boldsymbol{A} - \boldsymbol{B}_0\boldsymbol{L}_0 - \boldsymbol{F}_0\boldsymbol{C}_y & \boldsymbol{B}_0 & \boldsymbol{B}_0\boldsymbol{L}_0 & -\boldsymbol{B}_0 & \boldsymbol{F}_0 & \boldsymbol{B}_0 \\ \mathbf{0} & \mathbf{0} & r\boldsymbol{L}_0 & -r\boldsymbol{I} & \boldsymbol{L}_{I0} & r\boldsymbol{I} \\ -\boldsymbol{F}_0\boldsymbol{C}_y & \mathbf{0} & \boldsymbol{A}_0 & \mathbf{0} & \boldsymbol{F}_0 & \boldsymbol{B}_0 \\ -\boldsymbol{L}_{I0}\boldsymbol{C}_y & \mathbf{0} & \boldsymbol{L}_{I0}\boldsymbol{C}_y + r\boldsymbol{L}_0 & -r\boldsymbol{I} & \boldsymbol{L}_{I0} & r\boldsymbol{I} \\ \hline -\boldsymbol{L}_0 & \boldsymbol{I} & \boldsymbol{L}_0 & -\boldsymbol{I} & \mathbf{0} & \boldsymbol{I} \\ -\boldsymbol{C}_y & \mathbf{0} & \boldsymbol{C}_y & \mathbf{0} & \boldsymbol{I} & \mathbf{0} \end{array}\right] . \tag{D.27}$$

Appendix E
Observer gain matrices

E.1 Observer feedback gain matrices

$$\boldsymbol{F}_{0,1-6} = \begin{bmatrix} 9.98\cdot10^{-1} & 0 & 0 & 1.14\cdot10^{-11} & 8.10\cdot10^{-7} & 0 \\ 0 & 9.98\cdot10^{-1} & 0 & -8.10\cdot10^{-7} & 1.35\cdot10^{-11} & 1.28\cdot10^{-11} \\ 0 & 0 & 9.98\cdot10^{-1} & -3.64\cdot10^{-10} & 0 & 0 \\ 1.14\cdot10^{-11} & -8.10\cdot10^{-7} & -3.64\cdot10^{-10} & 9.98\cdot10^{-1} & 1.24\cdot10^{-10} & -2.59\cdot10^{-9} \\ 8.10\cdot10^{-7} & 1.35\cdot10^{-11} & 0 & 1.24\cdot10^{-10} & 9.98\cdot10^{-1} & 3.50\cdot10^{-10} \\ 0 & 1.28\cdot10^{-11} & 0 & -2.59\cdot10^{-9} & 3.50\cdot10^{-10} & 9.98\cdot10^{-1} \\ 2.49\cdot10^{-5} & 2.29\cdot10^{-10} & 5.05\cdot10^{-8} & 1.60\cdot10^{-8} & -2.43\cdot10^{-4} & 3.32\cdot10^{-9} \\ -2.23\cdot10^{-9} & 2.49\cdot10^{-5} & 9.81\cdot10^{-7} & 2.43\cdot10^{-4} & 3.74\cdot10^{-8} & -3.45\cdot10^{-8} \\ -5.04\cdot10^{-8} & -9.81\cdot10^{-7} & 2.49\cdot10^{-5} & -9.78\cdot10^{-6} & 4.88\cdot10^{-7} & 6.24\cdot10^{-9} \\ -1.41\cdot10^{-8} & 5.68\cdot10^{-8} & 1.06\cdot10^{-8} & 2.78\cdot10^{-5} & 5.59\cdot10^{-7} & 1.11\cdot10^{-7} \\ -8.03\cdot10^{-8} & -1.28\cdot10^{-8} & 1.50\cdot10^{-9} & -5.50\cdot10^{-7} & 2.76\cdot10^{-5} & 9.87\cdot10^{-7} \\ -1.83\cdot10^{-11} & 4.26\cdot10^{-8} & -5.51\cdot10^{-9} & -1.45\cdot10^{-6} & -9.96\cdot10^{-7} & 2.48\cdot10^{-5} \end{bmatrix} \tag{E.1}$$

$$\boldsymbol{F}_{0,7-12} = \begin{bmatrix} 2.49\cdot10^{-5} & -2.23\cdot10^{-9} & -5.04\cdot10^{-8} & -1.41\cdot10^{-8} & -8.03\cdot10^{-8} & -1.83\cdot10^{-11} \\ 2.29\cdot10^{-10} & 2.49\cdot10^{-5} & -9.81\cdot10^{-7} & 5.68\cdot10^{-8} & -1.28\cdot10^{-8} & 4.26\cdot10^{-8} \\ 5.05\cdot10^{-8} & 9.81\cdot10^{-7} & 2.49\cdot10^{-5} & 1.06\cdot10^{-8} & 1.50\cdot10^{-9} & -5.51\cdot10^{-9} \\ 1.60\cdot10^{-8} & 2.43\cdot10^{-4} & -9.78\cdot10^{-6} & 2.78\cdot10^{-5} & -5.50\cdot10^{-7} & -1.45\cdot10^{-6} \\ -2.43\cdot10^{-4} & 3.74\cdot10^{-8} & 4.88\cdot10^{-7} & 5.59\cdot10^{-7} & 2.76\cdot10^{-5} & -9.96\cdot10^{-7} \\ 3.32\cdot10^{-9} & -3.45\cdot10^{-8} & 6.24\cdot10^{-9} & 1.11\cdot10^{-7} & 9.87\cdot10^{-7} & 2.48\cdot10^{-5} \\ 9.98\cdot10^{-1} & 2.02\cdot10^{-9} & -1.46\cdot10^{-7} & 3.13\cdot10^{-6} & 3.11\cdot10^{-5} & -6.19\cdot10^{-8} \\ 2.02\cdot10^{-9} & 9.98\cdot10^{-1} & 5.64\cdot10^{-7} & -3.85\cdot10^{-5} & 3.55\cdot10^{-6} & 2.66\cdot10^{-5} \\ -1.46\cdot10^{-7} & 5.64\cdot10^{-7} & 9.98\cdot10^{-1} & 2.95\cdot10^{-6} & -2.37\cdot10^{-7} & -2.16\cdot10^{-6} \\ 3.13\cdot10^{-6} & -3.85\cdot10^{-5} & 2.95\cdot10^{-6} & 9.97\cdot10^{-1} & 2.52\cdot10^{-5} & 1.72\cdot10^{-5} \\ 3.11\cdot10^{-5} & 3.55\cdot10^{-6} & -2.37\cdot10^{-7} & 2.52\cdot10^{-5} & 9.97\cdot10^{-1} & -1.27\cdot10^{-7} \\ -6.19\cdot10^{-8} & 2.66\cdot10^{-5} & -2.16\cdot10^{-6} & 1.72\cdot10^{-5} & -1.27\cdot10^{-7} & 9.97\cdot10^{-1} \end{bmatrix} \tag{E.2}$$

$$\boldsymbol{F}_{1,1-6} = \begin{bmatrix} 9.98\cdot10^{-1} & 0 & 1.92\cdot10^{-11} & 7.83\cdot10^{-11} & 8.10\cdot10^{-7} & 0 \\ 0 & 9.98\cdot10^{-1} & 0 & 5.98\cdot10^{-6} & -2.58\cdot10^{-11} & 2.48\cdot10^{-4} \\ 1.92\cdot10^{-11} & 0 & 9.98\cdot10^{-1} & -6.04\cdot10^{-10} & -2.47\cdot10^{-4} & -1.95\cdot10^{-11} \\ 7.83\cdot10^{-11} & 5.98\cdot10^{-6} & -6.04\cdot10^{-10} & 9.97\cdot10^{-1} & 1.81\cdot10^{-9} & -5.22\cdot10^{-7} \\ 8.10\cdot10^{-7} & -2.58\cdot10^{-11} & -2.47\cdot10^{-4} & 1.81\cdot10^{-9} & 9.97\cdot10^{-1} & 2.90\cdot10^{-10} \\ 0 & 2.48\cdot10^{-4} & -1.95\cdot10^{-11} & -5.22\cdot10^{-7} & 2.90\cdot10^{-10} & 9.97\cdot10^{-1} \\ 2.49\cdot10^{-5} & 9.59\cdot10^{-11} & 8.88\cdot10^{-6} & 9.51\cdot10^{-9} & -2.43\cdot10^{-4} & -2.44\cdot10^{-8} \\ -2.00\cdot10^{-8} & 2.50\cdot10^{-5} & 6.58\cdot10^{-7} & 2.43\cdot10^{-4} & 2.63\cdot10^{-7} & -3.41\cdot10^{-6} \\ -7.11\cdot10^{-7} & -6.71\cdot10^{-7} & 2.45\cdot10^{-5} & -6.75\cdot10^{-6} & 1.02\cdot10^{-5} & 7.28\cdot10^{-8} \\ -1.39\cdot10^{-8} & 5.93\cdot10^{-8} & -4.94\cdot10^{-9} & 2.87\cdot10^{-5} & 4.72\cdot10^{-7} & 8.70\cdot10^{-8} \\ -8.00\cdot10^{-8} & -4.65\cdot10^{-9} & -2.39\cdot10^{-7} & 3.44\cdot10^{-7} & 2.75\cdot10^{-5} & 7.00\cdot10^{-7} \\ 2.94\cdot10^{-8} & 1.08\cdot10^{-7} & 2.76\cdot10^{-8} & 2.16\cdot10^{-5} & -4.27\cdot10^{-7} & 2.48\cdot10^{-5} \end{bmatrix} \tag{E.3}$$

$$\boldsymbol{F}_{1,7-12} = \begin{bmatrix} 2.49\cdot10^{-5} & -2.00\cdot10^{-8} & -7.11\cdot10^{-7} & -1.39\cdot10^{-8} & -8.00\cdot10^{-8} & 2.94\cdot10^{-8} \\ 9.59\cdot10^{-11} & 2.50\cdot10^{-5} & -6.71\cdot10^{-7} & 5.93\cdot10^{-8} & -4.65\cdot10^{-9} & 1.08\cdot10^{-7} \\ 8.88\cdot10^{-6} & 6.58\cdot10^{-7} & 2.45\cdot10^{-5} & -4.94\cdot10^{-9} & -2.39\cdot10^{-7} & 2.76\cdot10^{-8} \\ 9.51\cdot10^{-9} & 2.43\cdot10^{-4} & -6.75\cdot10^{-6} & 2.87\cdot10^{-5} & 3.44\cdot10^{-7} & 2.16\cdot10^{-5} \\ -2.43\cdot10^{-4} & 2.63\cdot10^{-7} & 1.02\cdot10^{-5} & 4.72\cdot10^{-7} & 2.75\cdot10^{-5} & -4.27\cdot10^{-7} \\ -2.44\cdot10^{-8} & -3.41\cdot10^{-6} & 7.28\cdot10^{-8} & 8.70\cdot10^{-8} & 7.00\cdot10^{-7} & 2.48\cdot10^{-5} \\ 9.98\cdot10^{-1} & -2.09\cdot10^{-7} & -6.90\cdot10^{-6} & 2.53\cdot10^{-6} & 3.92\cdot10^{-5} & 8.85\cdot10^{-6} \\ -2.09\cdot10^{-7} & 9.98\cdot10^{-1} & 1.08\cdot10^{-6} & -4.92\cdot10^{-5} & 2.68\cdot10^{-6} & -2.05\cdot10^{-4} \\ -6.90\cdot10^{-6} & 1.08\cdot10^{-6} & 9.97\cdot10^{-1} & 1.15\cdot10^{-5} & 3.01\cdot10^{-4} & 6.78\cdot10^{-6} \\ 2.53\cdot10^{-6} & -4.92\cdot10^{-5} & 1.15\cdot10^{-5} & 9.97\cdot10^{-1} & 1.87\cdot10^{-5} & 2.00\cdot10^{-5} \\ 3.92\cdot10^{-5} & 2.68\cdot10^{-6} & 3.01\cdot10^{-4} & 1.87\cdot10^{-5} & 9.97\cdot10^{-1} & -1.16\cdot10^{-5} \\ 8.85\cdot10^{-6} & -2.05\cdot10^{-4} & 6.78\cdot10^{-6} & 2.00\cdot10^{-5} & -1.16\cdot10^{-5} & 9.97\cdot10^{-1} \end{bmatrix} \tag{E.4}$$

E.2 LQR gain matrices

$$\boldsymbol{L}_0 = \begin{bmatrix} 1.4 \cdot 10^{-3} & 1.37 \cdot 10^{-3} & -1.9 \cdot 10^{-2} & -4.28 \cdot 10^{-7} & -2.51 \cdot 10^{-4} & -1.38 \cdot 10^{-3} & 1.28 \cdot 10^{-3} & 1.23 \cdot 10^{-4} & -1.41 \cdot 10^{-2} \\ -1.26 \cdot 10^{-2} & 5.15 \cdot 10^{-4} & 4.31 \cdot 10^{-5} & 5.47 \cdot 10^{-6} & 6.32 \cdot 10^{-5} & -2.57 \cdot 10^{-5} & -1.11 \cdot 10^{-2} & -7.85 \cdot 10^{-5} & -1.58 \cdot 10^{-4} \\ -2.65 \cdot 10^{-4} & -1.7 \cdot 10^{-2} & -4.08 \cdot 10^{-4} & -7.37 \cdot 10^{-6} & 3.89 \cdot 10^{-6} & -2.04 \cdot 10^{-6} & -1.62 \cdot 10^{-4} & -7.47 \cdot 10^{-3} & 1.7 \cdot 10^{-6} \\ -9.43 \cdot 10^{-5} & -7.53 \cdot 10^{-5} & 9.16 \cdot 10^{-4} & 7.2 \cdot 10^{-9} & 1.16 \cdot 10^{-5} & -2.77 \cdot 10^{-6} & -8.57 \cdot 10^{-5} & -1.07 \cdot 10^{-5} & 6.78 \cdot 10^{-4} \end{bmatrix} \tag{E.5}$$

$$\boldsymbol{L}_{0I} = \begin{bmatrix} -3.33 \cdot 10^{-4} & -3.05 \cdot 10^{-4} & 4.20 \cdot 10^{-3} & 8.96 \cdot 10^{-8} & -9.75 \cdot 10^{-7} & 3.94 \cdot 10^{-4} & -4.50 \cdot 10^{-4} & -1.94 \cdot 10^{-5} & 4.17 \cdot 10^{-3} \\ 2.86 \cdot 10^{-3} & -9.98 \cdot 10^{-5} & -6.85 \cdot 10^{-6} & 1.09 \cdot 10^{-7} & 6.89 \cdot 10^{-6} & 7.33 \cdot 10^{-7} & 4.08 \cdot 10^{-3} & -7.47 \cdot 10^{-5} & -7.96 \cdot 10^{-6} \\ 5.34 \cdot 10^{-5} & 3.78 \cdot 10^{-3} & 9.11 \cdot 10^{-5} & -3.43 \cdot 10^{-6} & 1.38 \cdot 10^{-7} & -2.72 \cdot 10^{-9} & 9.16 \cdot 10^{-5} & 2.29 \cdot 10^{-3} & -1.76 \cdot 10^{-6} \\ 2.22 \cdot 10^{-5} & 1.68 \cdot 10^{-5} & -2.02 \cdot 10^{-4} & 1.05 \cdot 10^{-9} & 2.40 \cdot 10^{-7} & 3.53 \cdot 10^{-7} & 3.06 \cdot 10^{-5} & 2.20 \cdot 10^{-6} & -2.01 \cdot 10^{-4} \end{bmatrix} \tag{E.6}$$

$$\boldsymbol{L}_1 = \begin{bmatrix} 7.49 \cdot 10^{-4} & 1.75 \cdot 10^{-3} & -2.46 \cdot 10^{-2} & 2.64 \cdot 10^{-4} & 6.50 \cdot 10^{-4} & -3.69 \cdot 10^{-2} & 3.54 \cdot 10^{-5} & -1.23 \cdot 10^{-2} & -1.49 \cdot 10^{-3} \\ -1.68 \cdot 10^{-1} & 5.74 \cdot 10^{-2} & 8.39 \cdot 10^{-3} & -3.98 \cdot 10^{-3} & -2.23 \cdot 10^{-2} & 1.17 \cdot 10^{-2} & -5.36 \cdot 10^{-2} & 1.01 \cdot 10^{-2} & 6.30 \cdot 10^{-3} \\ -5.67 \cdot 10^{-4} & -1.44 \cdot 10^{-1} & -1.97 \cdot 10^{-2} & 7.98 \cdot 10^{-3} & 1.11 \cdot 10^{-4} & -7.26 \cdot 10^{-3} & 1.40 \cdot 10^{-4} & -2.37 \cdot 10^{-2} & -8.06 \cdot 10^{-4} \\ -4.17 \cdot 10^{-2} & -2.65 \cdot 10^{-2} & 1.84 & 1.51 \cdot 10^{-3} & -5.72 \cdot 10^{-2} & -6.98 \cdot 10^{-3} & -2.57 \cdot 10^{-6} & -3.79 \cdot 10^{-3} & 1.14 \cdot 10^{-1} \end{bmatrix} \tag{E.7}$$

$$\boldsymbol{L}_{1I} = \begin{bmatrix} -3.94 \cdot 10^{-7} & -2.83 \cdot 10^{-4} & 2.85 \cdot 10^{-3} & 1.40 \cdot 10^{-5} & -6.27 \cdot 10^{-5} & 9.18 \cdot 10^{-3} & 5.52 \cdot 10^{-7} & -4.52 \cdot 10^{-5} & 6.82 \cdot 10^{-6} \\ 3.78 \cdot 10^{-3} & -6.37 \cdot 10^{-3} & -9.35 \cdot 10^{-4} & 4.83 \cdot 10^{-4} & 3.00 \cdot 10^{-3} & -2.34 \cdot 10^{-3} & 5.04 \cdot 10^{-3} & -1.43 \cdot 10^{-3} & -1.42 \cdot 10^{-6} \\ 1.63 \cdot 10^{-5} & 1.82 \cdot 10^{-2} & 2.39 \cdot 10^{-3} & -1.12 \cdot 10^{-3} & -1.10 \cdot 10^{-5} & -1.25 \cdot 10^{-5} & 2.56 \cdot 10^{-5} & 3.46 \cdot 10^{-3} & 3.04 \cdot 10^{-6} \\ 1.85 \cdot 10^{-4} & 3.37 \cdot 10^{-3} & -2.13 \cdot 10^{-1} & 2.03 \cdot 10^{-5} & 5.47 \cdot 10^{-3} & 1.89 \cdot 10^{-3} & 1.74 \cdot 10^{-4} & 9.01 \cdot 10^{-5} & -5.09 \cdot 10^{-4} \end{bmatrix} \tag{E.8}$$

Appendix F
Supervisor parameter determination

This appendix describes the determination of the supervisor parameters velocity rate limit, maximum velocity reference and the deceleration radius from a waypoint. These parameters are determined based on several iterative test simulations, and are performed for each of the three controllers; hover, forward flight and gain scheduling.

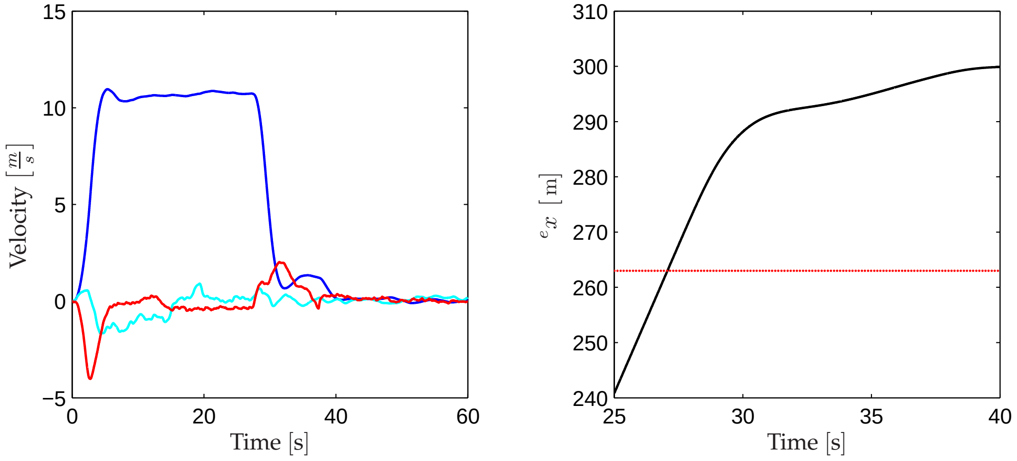
F.1 Test description

The supervisor is given a target waypoint such that the UAV must travel a distance of $300\,\mathrm{m}$ in a straight line from its starting point. Furthermore, the UAV is initialized in the hover operating point, which results in an acceleration at the start of the test and a deceleration when the waypoint is reached. It is then possible to determine the velocity rate limit $v_{r,\text{rate limit}}$ and the deceleration radius $R_{v_{r,\text{slow}}}$. The distance $R_{v_{r,\text{slow}}}$ is determined as the minimum distance the controller needs to decelerate to $v_{r,\text{slow}}$ before entering the waypoint radius $R_{v_{r,\text{hover}}} = 2\,\mathrm{m}$. Furthermore, the maximum velocity reference $v_{r,\text{fast}}$ is determined such that the highest possible forward flight velocity is obtained with a maximum overshoot of $2\,\frac{\mathrm{m}}{\mathrm{s}}$, and a maximum of lateral velocity overshoot of $2\,\frac{\mathrm{m}}{\mathrm{s}}$.

F.2 Determination of parameters

The supervisor parameters with respect to the hover controller are determined using the above described procedure. From iterative test simulations the maximum velocity rate limit for the hover controller is identified as $5\frac{\mathrm{m}}{\mathrm{s}^2}$. In addition, the maximum velocity obtained for the hover controller, with the above stated requirements, is determined as $11\,\frac{\mathrm{m}}{\mathrm{s}}$ (see Figure F.1(a)). Based on the maximum velocity and the deceleration rate the deceleration radius for the hover controller is identified as $37\,\mathrm{m}$ (see Figure F.1(b)). The supervisor parameters regarding the forward flight controller are determined in the same manner as for the hover controller, where the test results are depicted in Figure F.2(a) and F.2(b). From the iterative tests for the forward flight controller the maximum velocity rate limit is identified as $5\frac{\mathrm{m}}{\mathrm{s}^2}$. Furthermore, the maximum velocity obtained for the forward flight controller is determined as $15\,\frac{\mathrm{m}}{\mathrm{s}}$, and the deceleration distance to the waypoint is identified as $60\,\mathrm{m}$. It can be determined, that the forward flight controller is more useful regarding completion time of level 1 of the IARC, because of the higher maximum velocity.

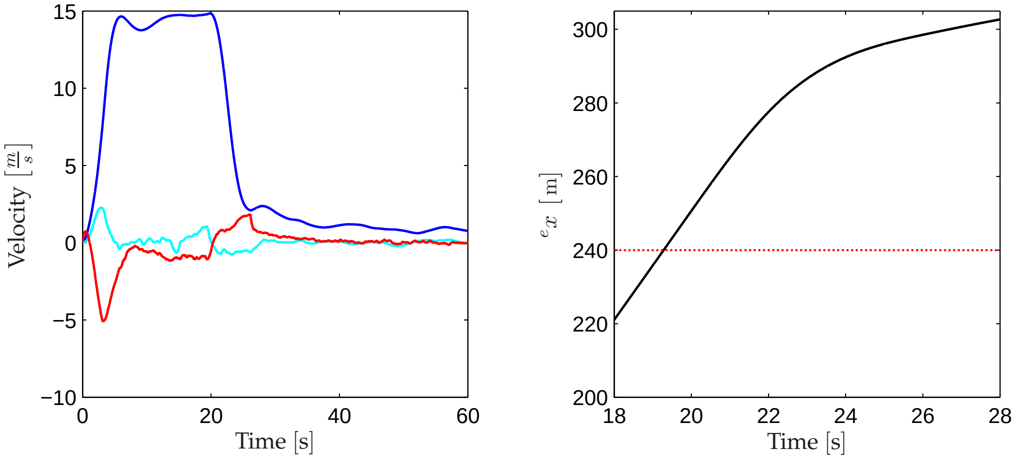
The last supervisor parameters to be determined are the parameters regarding the gain scheduling controller, where the test results are illustrated in Figure F.3(a) and F.3(b). In the tests the



(a)

(b)

FIGURE F.1: *Graphs for determination of supervisor parameters for the hover controller. (a) is used for determination of $v_{r,fast}$ and $v_{r,rate\ limit}$; blue is ${}^b\dot{x}$, cyan is ${}^b\dot{y}$ and red is ${}^b\dot{z}$. (b) is used for determination of $R_{v_{r,slow}}$ represented by the red dashed line.*



(a)

(b)

FIGURE F.2: *Graphs for determination of supervisor parameters for the forward flight controller. (a) is used for determination of $v_{r,fast}$ and $v_{r,rate\ limit}$; blue is ${}^b\dot{x}$, cyan is ${}^b\dot{y}$ and red is ${}^b\dot{z}$. (b) is used for determination of $R_{v_{r,slow}}$ represented by the red dashed line.*

maximum velocity rate limit is determined as $5\,\frac{\mathrm{m}}{\mathrm{S}^2}$, and the maximum velocity obtained for the gain scheduling controller is identified as $16\,\frac{\mathrm{m}}{\mathrm{s}}$. Based on the velocity rate limit and maximum velocity of the gain scheduling controller, the deceleration radius is identified as $62\,\mathrm{m}$ from the target waypoint.

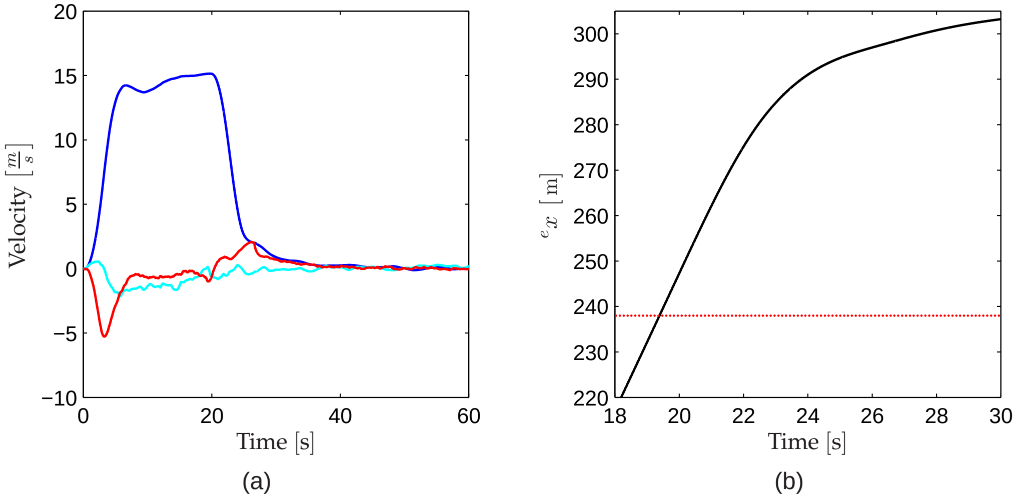


FIGURE F.3: *Graphs for determination of supervisor parameters for the gain scheduling controller. (a) is used for determination of $v_{r,fast}$ and $v_{r,rate\ limit}$; blue is ${}^b\dot{x}$, cyan is ${}^b\dot{y}$ and red is ${}^b\dot{z}$. (b) is used for determination of $R_{v_{r,slow}}$ represented by the red dashed line.*

The results of the three experiments are listed in Table F.1

Parameter	Hover	Forward flight	Gain scheduling
$v_{r,\text{fast}}$	$11\,\frac{\text{m}}{\text{s}}$	$15\,\frac{\text{m}}{\text{s}}$	$16\,\frac{\text{m}}{\text{s}}$
$v_{r,\text{rate limit}}$	$\pm5\frac{\text{m}}{\text{s}^2}$	$\pm5\frac{\text{m}}{\text{s}^2}$	$\pm5\frac{\text{m}}{\text{s}^2}$
$R_{v_{r,\text{slow}}}$	$37\,\text{m}$	$60\,\text{m}$	$62\,\text{m}$

TABLE F.1: *List of the determined supervisor parameters for the three controllers; hover, forward flight and gain scheduling.*