University of Stavanger

Faculty of Science and Technology

MASTER'S THESIS

Study program/ Specialization:	Spring semester, 2022. open Open / Restricted access
Writer: Emil Kleivdal	Emil Eikanger Kleival …… (Writer's signature)
Faculty supervisor: Damiano Rotondo External supervisor(s):	
Thesis title: Sampled-data strategies for the control of a 3 DOF hover system	
Credits (ECTS):	
Key words:	Pages: 57 …… + enclosure: 33 …… Stavanger, 2022 …….. Date/year

Front page for master thesis
Faculty of Science and Technology
Decision made by the Dean October 30th 2009

University of Stavanger

Faculty of Science and Technology
Department of Electrical Engineering and Computer Science

Sampled-data strategies for the control of a 3 DOF hover system

Master's Thesis in Robot Technology and Signal Processing
by

Emil Eikanger Kleivdal

Internal Supervisors

Damiano Rotondo

June 15, 2022

Abstract

This thesis will investigate different sampled-data strategies that can be used for the digital implementation of a controller. To implement the controller several discretization techniques are considered and compared including implicit and explicit Euler methods, Tustin approximation and zero-order hold. The first sampled-data strategy to be investigated is classical periodic sampling, where the output signal is measured periodically, and the controller output is updated every fixed amount of time. The deterioration of performance as sampling times increase as well as the stability properties is investigated for the different discretized models. In a more advanced stage of this project, an event-triggered strategy is considered. In event-triggered control, sensor sampling and control updates are generated only when the system's state deviate too much from the desired value, which offers a reactive approach to control.

Acknowledgements

I would like to thank my facculty supervisor Assoc. Prof. Damiano Rotondo for his invaluable help and feedback throughout this thesis. Without his insigth and patience i would not have made it through my masters degree. Thank you!

Contents

Abbreviations

NCS	**N**etworked **C**ontrol **S**ystem
DOF	**D**egrees **O**f **F**reedom
CCW	**C**ounter **C**lockwise
CW	**C**lockwise
ODE	**O**rdinary **D**ifferential **E**quation
MIMO	**M**ultiple **I**nput **M**ultiple **O**utput
SISO	**S**ingle **I**nput **S**ingle **O**utput
SDC	**S**tate **D**ependent **C**oefficient
CT	**C**ontinuous **T**ime
DT	**D**iscreet **T**ime
FE	**F**orward **E**uler
BE	**B**ackward **E**uler
TU	**T**ustin **A**pproximation
ZOH	**Z**ero **O**rder **H**old
D/A	**D**igital **A**nalog
LTI	**L**inear **T**ime **I**nvariant
LQR	**L**inear **Q**uadratic **R**egulator
DARE	**D**iscrete-time **A**lgebraic **R**iccati **E**quation
ZOH	**Z**ero **O**rder **H**old

Chapter 1

Motivation

1.1 Introduction

One of the most fundamental concepts of system theory is the combining of multiple simple interconnecting simple systems in order to create more powerful and complex ones. Developments in computer and communication technologies during recent years have led to an advance in new types of large-scale embedded and networked control systems (NCS). In these systems sensors, controllers and actuators are connected by a network or other shared framework. These systems operate under communication constraints and as such it is desirable to limit the both the required monitoring of sensors and the amount of computation instants to just the instances in which the system requires attention. In a lot of today's standard courses in control theory, periodic sampling is considered the best approach for implementing digital feedback control laws. Dealing with constrained resources can simply be the utilization of efficient, robust sampling with proper discretized models, however alternatives in the form of aperiodic control, often referred to as event-based control has been around since as early as the 1960's. The discussion of periodic versus aperiodic control has since come in and out of fashion, but in the early 2000's as the popularity of NCS's has increased and papers detailing the advantages and viability of event-based control started appearing the discussion has made a resurgence.

1.2 History of event-based and resource efficient control

Periodic sampling is not required. The most suitable sampling is by transmission of only significant data, as the new value obtained when the data are changed by a given

increment. In certain cases, transmission of data by this means can be used to increase channel capacity. -P.H. Ellis, 1959

The first mention of event-based control appeared in 1959 by research engineer P. H. Ellis as digital control had only just begun to emerge. Since then it has been applied in several fields in the last 63 years, some notable mentions are:

- In **1963 S. Gupta** [1] applied event-based sampling to increase the efficiency of control systems
- In **1966 A. Liff and J. Wolf** [2] discussed the optimal sampling rate for discrete-time modelling of continuous-time systems.
- In **1966 G. Bekey and R. Tomovic** [3] wrote about the sensitivity of discrete systems to variation of sampling interval.
- In **1966 G. Bekey and R. Tomovic** [4] produced a second article the same year discussing adaptive sampling based on amplitude sensitivity.
- In **1967 D. Ciscato and L. Martiani** [5] wrote an article on increasing sampling efficiency by adaptive sampling.
- In **1969 J. Mitchell and J. McDaniel** [6] explored an adaptive sampling technique.

1.3 Objective

The main objectives of this thesis is to investigate sampled-data strategies for control of a continuous system. The steps taken can be segmented into the following parts:

- Develop a state-space representation of the 3DOF hover.
- Develop several discretized approximations of the state-space model.
- Compare the different models and come to a conclusion about the performance of the different discretization methods.
- Develop a digital feedback control law for the system based on the discretized models.
- Construct a functional event-triggering condition for efficient sampling of the system.

The different sampled-data strategies will be tested on the 3DOF Hover produced by Quanser, see Figure 2.1 which is available in the laboratory located in KE-E458.

Chapter 2

Modelling the system

The theoretical results described in this report are applied to the 3 degrees of freedom (DOF) Hover located at the University of Stavanger in room KE E-455. Section 2.1 describes the physical system and its components. Assuming a model linearized at the equilibrium point in which the propellers are aligned with the pitch, roll and yaw axis, equations (2.2), (2.3) and (2.5) in subsections 2.1.1 and 2.1.2 describes motion about the body axes of the system. Section 2.2 details the construction of the same linear state-space model as is provided by the manufacturers manual, [7]. In section 2.2.1 it is demonstrated how the same model can be acquired from a nonlinear model based on Newton-Euler formulations and Jacobian linearization. In section 2.3 several methods for discretization of the state-space model are compared.

2.1 The Quanser 3DOF Hover

The Hover consists of four propellers connected by a round, planar frame. Each propeller is driven by its own DC motor mounted to the frame, the frame is in turn mounted to a three DOF joint allowing the body to freely move about the three axes. The front and back propellers rotate counter-clockwise (CCW), while the left and right propellers rotate clockwise (CW). This ensures that when the thrust from the four propellers are equal the total torque of the system is balanced. The propellers generate lift to control the roll and pitch angles and the motion about the yaw axis is produced by the total difference in torques generated by the CW propeller pair and the CCW pair. The hover has three encoders that measure the angles, each with a resolution of 8192 counts per revolution, thus the precision of the measured angles is accurate down to 0.0439 degrees.

Figure 2.1: 3DOF Hover from Quanser. [7]

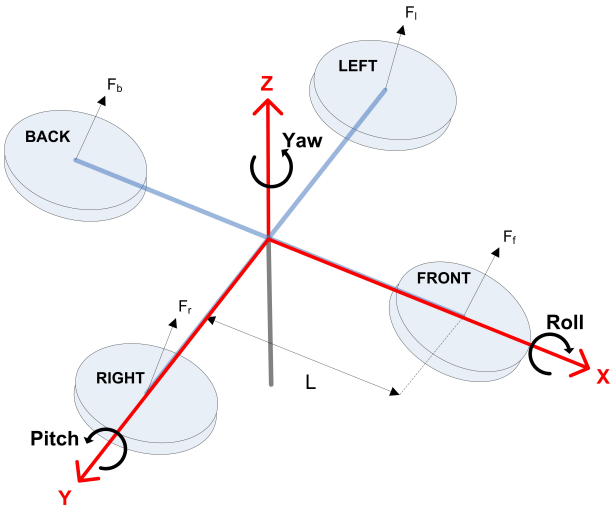


Figure 2.2: Free body diagram of the hover. [7]

The thrust force generated by the front, back left and right motors is denoted by F_f, F_b, F_l and F_r. The pitch, roll and yaw angles are denoted by θ_p, θ_r and θ_y, respectively. When $\theta_p = \theta_r = 0$ the hover is parallel to the ground.

2.1.1 Pitch and Roll axis model

Rotation about the pitch axis can be described by the equation (2.1)

$$J_p \ddot{\theta}_p = \Delta F L \tag{2.1}$$

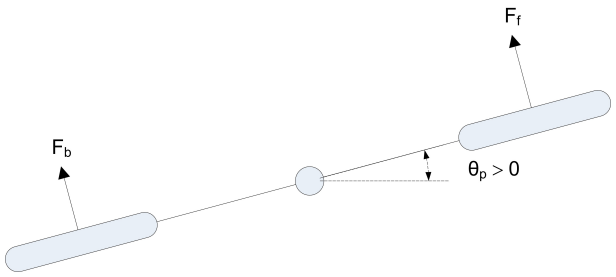


Figure 2.3: Free body diagram of the pitch axis. [7]

where J_p is the moment of inertia about the pitch axis, ΔF is the thrust-force differential, L is the distance between the propeller and the pivot point and θ_p is the pitch angle. From Fig 2.3 it is clear that θ_p increases as F_f increases in relation to F_b. The force generated by a motor/propeller pair is assumed to be $K_f V$, resulting in $\Delta F = K_f(V_f - V_b)$. Using Equation (2.1) a model for the pitch axis can be derived as:

$$J_p \ddot{\theta}_p = LK_f(V_f - V_b) \tag{2.2}$$

where the meaning of the different parameters is described in table 2.1.

Symbol	Description	Value	Unit
J_p	Moment of inertia about pitch axis	0.0552	$Kg - m^2$
L	Distance from propeller to pivot point	0.197	m
K_f	Force-thrust constant of motor/propeller	0.1188	N/V
θ_p	Pitch angle	–	Rad
V_f	Voltage applied to front motor	–	V
V_b	Voltage applied to back motor	–	V

Table 2.1: Variables regarding pitch model.

The exact same can be shown for the roll axis:

$$J_r \ddot{\theta}_r = LK_f(V_r - V_l) \tag{2.3}$$

an overview of the different parameters are listed in table 2.2.

2.1.2 Yaw axis model

Rotation around the yaw axis is as mentioned caused by the total difference in torques generated by the CW propeller pair and CCW pair. The torque generated by each propeller is assumed to be $\tau = K_t V_m$[7] where V_m is the motor voltage.

$$J_y \ddot{\theta}_y = \Delta\tau = \tau_l + \tau_r - \tau_f - \tau_b \tag{2.4}$$

Symbol	Description	Value	Unit
J_r	Moment of inertia about roll axis	0.0552	$Kg - m^2$
L	Distance from propeller to pivot point	0.197	m
K_f	Force-thrust constant of motor/propeller	0.1188	N/V
θ_r	Roll angle	–	Rad
V_r	Voltage applied to right motor	–	V
V_l	Voltage applied to left motor	–	V

Table 2.2: Variables regarding roll model.

where τ_l and τ_r is the CW pair and τ_f and τ_b is the CCW pair. In terms of motor voltages Equation (2.4) becomes:

$$J_y\ddot{\theta}_y = K_t(V_l + V_r) - K_t(V_f + V_b) \tag{2.5}$$

parameters for the yaw model are listed in table 2.3.

Symbol	Description	Value	Unit
J_y	Moment of inertia about yaw axis	0.110	$Kg - m^2$
K_t	Torque thrust constant of motor/propeller	0.0036	$N - m/V$
θ_y	Yaw angle	–	Rad
V_r	Voltage applied to right motor	–	V
V_l	Voltage applied to left motor	–	V
V_f	Voltage applied to front motor	–	V
V_b	Voltage applied to back motor	–	V
τ_m	Torque generated by individual motor	K_tV_m	Nm

Table 2.3: Variables regarding yaw model.

2.2 State-space model

"Since Newton, mankind has come to realize that the laws of physics are always expressed in the language of differential equations." -Steven Strogatz

A dynamic model generally consists of one or more ordinary differential equations (ODEs). As a systems complexity grows it becomes increasingly challenging to express it well using transfer functions. In the case of MIMO systems like the 3DOF hover this is especially the case. A way to largely mitigate this issue is to use state-space modelling to represent the system. State-space modelling provides a practical, concise representation of a physical system. For linear systems the state-space representation replaces an n-th order ODE with a single matrix equation containing a set of p input, q output and n state variables, where the input and state variables are related by a set of ODEs.

The general state-space representation of a linear system is given by equations (2.6) and (2.7)

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{2.6}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{2.7}$$

Where **bold symbols** denote vectors and matrices and plain text represents scalars. Vectors and matrices used in the state-space model are listed in table 2.4.

Symbol	Description
$\mathbf{x} \in \mathbb{R}^n$	State vector
$\mathbf{y} \in \mathbb{R}^q$	Output vector
$\mathbf{u} \in \mathbb{R}^p$	Input vector
$\mathbf{A} \in \mathbb{R}^{n\times n}$	System matrix
$\mathbf{B} \in \mathbb{R}^{n\times p}$	Input matrix
$\mathbf{C} \in \mathbb{R}^{q\times n}$	Output matrix
$\mathbf{D} \in \mathbb{R}^{q\times n}$	Feedthrough matrix

Table 2.4: Notations regarding state-space representation.

In Section 2.1, expressions for motion about the pitch, roll and yaw axis' was derived. Taking Equations (2.2), (2.3) and (2.5) and solving for $\ddot{\theta}$ gives the basis for a state-space representation of the hover system:

$$\ddot{\theta}_p = \frac{LK_f}{J_p}(V_f - V_b) \tag{2.8}$$

$$\ddot{\theta}_r = \frac{LK_f}{J_r}(V_r - V_l) \tag{2.9}$$

$$\ddot{\theta}_y = \frac{K_t}{J_y}(V_r + V_l) + \frac{K_t}{J_y}(V_f + V_b) \tag{2.10}$$

The state variables should be chosen such that $\mathbf{x}$ describes the system at its current state, and that knowledge of the state variables at $t = t_0$ along with knowledge of the inputs at $t > t_0$ completely determines any future state of the system. A suitable choice of state variables are $\theta_p, \theta_r, \theta_y, \dot{\theta}_p, \dot{\theta}_r$ and $\dot{\theta}_y$ as it satisfies the above conditions. The output vector, $\mathbf{y}$ will be the angles and the input vector, $\mathbf{u}$ is the motor voltages.

$$\mathbf{x}^T = \begin{bmatrix} \theta_p & \theta_r & \theta_y & \dot{\theta}_p & \dot{\theta}_r & \dot{\theta}_y \end{bmatrix}$$

$$\mathbf{y}^T = \begin{bmatrix} \theta_p & \theta_r & \theta_y \end{bmatrix}$$

$$\mathbf{u}^T = \begin{bmatrix} V_f & V_b & V_l & V_r \end{bmatrix}$$

This results in the following state-space representation:

$$\dot{x}_1(t) = \dot{\theta}_p = x_4(t)$$

$$\dot{x}_2(t) = \dot{\theta}_r = x_5(t)$$

$$\dot{x}_3(t) = \dot{\theta}_y = x_6(t)$$

$$\dot{x}_4(t) = \ddot{\theta}_p = \frac{LK_f}{J_p}(u_1(t) - u_2(t))$$

$$\dot{x}_5(t) = \ddot{\theta}_r = \frac{LK_f}{J_r}(u_4(t) - u_3(t))$$

$$\dot{x}_6(t) = \ddot{\theta}_y = \frac{K_t}{J_y}(u_3(t) + u_4(t)) - \frac{K_t}{J_y}(u_1 + u_2)$$

$$y_1(t) = \theta_p = x_1(t)$$

$$y_2(t) = \theta_r = x_2(t)$$

$$y_3(t) = \theta_y = x_3(t)$$

Putting the vectors above into Equations (2.6) and (2.7) gives the compact state-space representation for the hover system.

$$\begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \\ \dot{x_3}(t) \\ \dot{x_4}(t) \\ \dot{x_5}(t) \\ \dot{x_6}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_6(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{LK_f}{J_p} & -\frac{LK_f}{J_p} & 0 & 0 \\ 0 & 0 & -\frac{LK_f}{J_r} & \frac{LK_f}{J_r} \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} \end{bmatrix}}_{\mathbf{B}} \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ u_4(t) \end{bmatrix} \tag{2.11}$$

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_6(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{D}} \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ u_4(t) \end{bmatrix}$$

2.2.1 Nonlinear model

Since the hover does in fact have nonlinear dynamics due to coupling between the axes it is worth looking into a nonlinear mathematical model of the system. In [8] a set of equations describing the dynamics of a rigid quad-copter with symmetrical axes was derived using Newton-Euler formulations.

$$\dot{\phi} = p + q \cdot sin(\phi)tan(\theta) + r \cdot cos(\phi)tan(\theta) \tag{2.12}$$

$$\dot{\theta} = q \cdot cos(\phi) - r \cdot sin(\phi) \tag{2.13}$$

$$\dot{\psi} = q \cdot sin(\phi)sec(\theta) + r \cdot cos(\phi)sec(\theta) \tag{2.14}$$

$$\dot{p} = \frac{J_p - J_y}{J_r} qr + \frac{1}{J_r} \tau_r \tag{2.15}$$

$$\dot{q} = \frac{J_y - J_r}{J_p} pr + \frac{1}{J_p} + \frac{1}{J_p}\tau_p \tag{2.16}$$

$$\dot{r} = \frac{J_r - J_p}{J_y} pq + \frac{1}{J_y}\tau_y \tag{2.17}$$

where ϕ, θ, ψ are the Euler angles and p, q, r are the angular rates. Furthermore J_r, J_p and J_y are the moments of inertia and τ_p, τ_r and τ_y are the torques as described by equations (2.2), (2.3) and (2.5). Choosing state variables as $x^T = \begin{bmatrix}\phi & \theta & \psi & p & q & r\end{bmatrix}$ and the input vector as $u^T = \begin{bmatrix}V_f & V_b & V_l & V_r\end{bmatrix}$ in combination with equations (2.12)-(2.17) result in the following state-dependent coefficient (SDC) state and input matrices $\mathbf{A}(x)$ and $\mathbf{B}(x)$.

$$\mathbf{A}(x) = \begin{bmatrix} 0 & 0 & 0 & 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & 0 & 0 & 0 & cos(\phi) & -sin(\phi) \\ 0 & 0 & 0 & 0 & \frac{sin(\phi)}{cos(\theta)} & \frac{cos(\phi)}{cos(\theta)} \\ 0 & 0 & 0 & 0 & 0 & \frac{J_p - J_y}{J_r} q \\ 0 & 0 & 0 & \frac{J_y - J_r}{J_p} r & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{J_r - J_p}{J_y} r & 0 \end{bmatrix}$$

$$\mathbf{B}(x) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{LK_f}{J_p} & -\frac{LK_f}{J_p} & 0 & 0 \\ 0 & 0 & \frac{LK_f}{J_r} & -\frac{LK_f}{J_r} \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} \end{bmatrix}$$

Linearizing the resulting matrices $\mathbf{A}(x)$ around the equilibrium point $x = 0$ using by Jacobian linearization yields:

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

whilst $\mathbf{B}(x)$ remain the same. The linear matrices obtained are indeed identical to the model provided by the manufacturer of the system.

2.3 Discretization

A sampled data control system refers to a control system in which a continuous plant interacts with a digital controller. Since any digital system operates in the discrete-time realm the reason for discretizing the continuous-time system are fairly obvious. It allows the controller to be implemented in the first place and to observe the estimated continuous-time system.

When designing a discrete-time controller for a continuous-time system there are two fundamental approaches to consider. One approach is to design a controller in the continuous-time domain and subsequently discretize it, ending up with a controller that approximates the behaviour of the original. The other approach and the one focused on in this thesis is to derive a discrete-time equivalent of the continuous system and directly design a discrete-time controller that satisfies the desired requirements for the feedback control system.

In the physical world signals are usually continuous, meaning data exist for any time t. In a computer however, everything is discrete. Before a continuous system can interact with a discrete-time controller, a discrete-time sequence must be derived from the original signal. This process is called sampling and can be done by passing a signal through an analog-digital (A/D) converter which transforms the continuous signal $y(t)$ into a discrete-time signal $y(k)$.

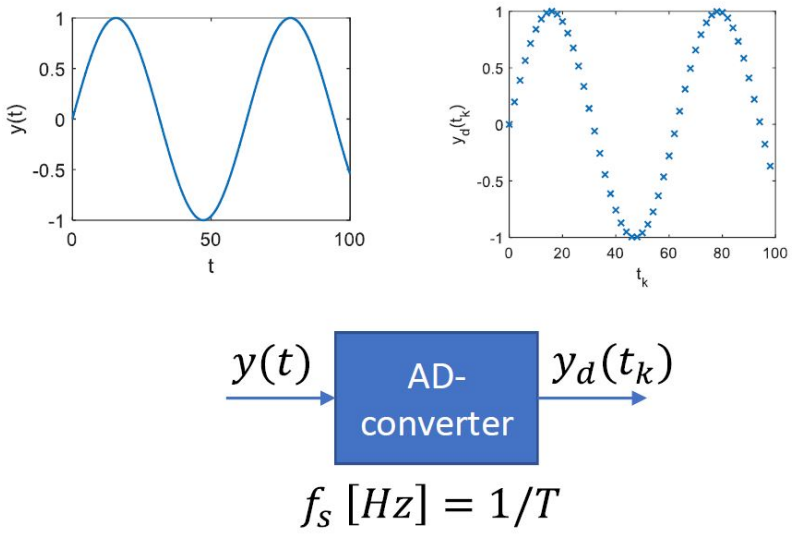


Figure 2.4: Sampling process by means of an A/D converter. Taken from [9]

Usually, but not always, the sampled signal is a sequence of equally spaced sampling instants $t_k = kT$, with k being a integer time index separated by a time interval T known as the sampling time. In the case of T being constant as illustrated in Fig. 2.4, it is

referred to as periodic sampling. Several sampling strategies are explored in this thesis, however in the following subsections calculations are made using periodic sampling.

In Section 2.2, it was shown that a linear, time-invariant continuous-time system can be represented by a set of first order matrix differential equations of the form (2.6) with $\dot{x}(t)$ being the continuous-time (CT) derivative of x. By first finding a discrete-time (DT) solution to the differential equation it is possible to obtain a DT state-space representation of the system in question. There are several ways to approximate this solution, the simplest of which are the forward and backward Euler's methods, Tustin approximation and Zero-order hold discretization.

2.3.1 Forward Euler

Considering the problem of computing an integral:

$$J = \int_0^t f(\tau)\, d\tau \tag{2.18}$$

The forward Euler method is the simplest approach to numerical integration of ODEs. The method is explicit in that it calculates the state of a system at a future time as a function of the current state of the system. The method works by approximating the integral of $f(t)$ by summing the area of each rectangle. The area is calculated taking the amplitude at kT and looking forward T units of time until the next sampling instance $(k+1)T$.

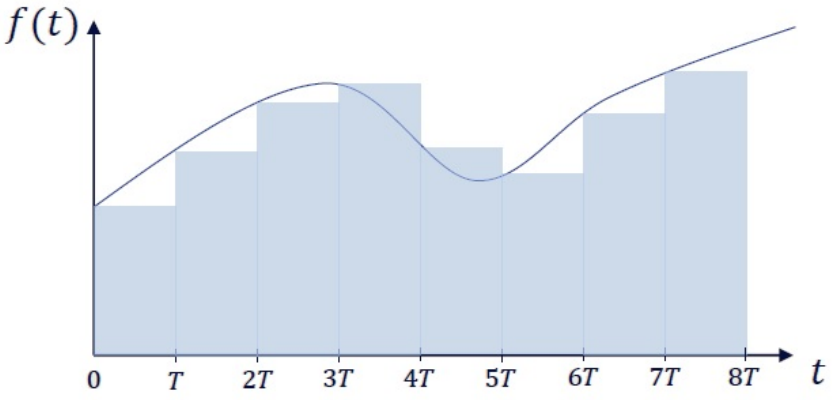


Figure 2.5: Forward Euler approximation of the integral of $f(t)$. Taken from [9]

The forward Euler method can be derived by applying this to the generic ODE (2.18) integrating over one time step of the differential equation from kT to $(k+1)T$ and

applying the fundamental theory of calculus

$$x((k+1)T) - x(kT) = \int_{kT}^{(k+1)T} f(\tau)\, d\tau \tag{2.19}$$

using the left Riemann sum the integral of one time step can be written as

$$\int_{kT}^{(k+1)T} f(\tau)\, d\tau \approx T f(x(kT)) \tag{2.20}$$

then by substituting for the integral, the next state of the system at time $(k+1)T$ can be approximated as

$$x((k+1)T) = x(kT) + T f(x(kT))$$

Where $x(kT)$ is the value of x at time $t_k = kT$ and $x((k+1)T)$ is the value of x at time $t_{k+1} = (k+1)T$.

Finally, if we look at the definition of the derivative:

$$\dot{x} = \lim_{\delta t \to 0} \frac{\delta x}{\delta t} \tag{2.21}$$

Where δx is the change in x over time interval δt. The forward finite difference approximation of the derivative (2.22) can be related to the forward Euler approximation.

$$\dot{x}(t) \cong \frac{x((k+1)T) - x(kT)}{T} = f(x(kT)) \tag{2.22}$$

Applying the forward Euler approximation to a continuous state-space model is achieved by substituting (2.22) into expression (2.6):

$$\frac{x((k+1)T) - x(kT)}{T} = \mathbf{A}x(kT) + \mathbf{B}u(kT)$$

$$x((k+1)T) - x(kT) = T\mathbf{A}x(kT) + T\mathbf{B}u(kT)$$

$$x((k+1)T) = (\mathbf{I} + T\mathbf{A})x(kT) + T\mathbf{B}u(kT)$$

Comparing equation (2.7) for the output vector $\mathbf{y}(\cdot)$ at time $t_k = kT$ with the general output vector for a continuous system, implies that $\mathbf{C}_d^{FE}$ and $\mathbf{D}_d^{FE}$ are unchanged from the original output and feedthrough matrices.

The discretized model is then given by:

$$x((k+1)T) = \mathbf{A}_d^{FE} x(kT) + \mathbf{B}_d^{FE} u(kT)$$

$$y(kT) = \mathbf{C}_d^{FE} x(kT) + \mathbf{D}_d^{FE} u(kT)$$

where:

$$\begin{aligned}
\mathbf{A}_d^{FE} &= \mathbf{I} + \mathbf{A} \\
\mathbf{B}_d^{FE} &= T\mathbf{B} \\
\mathbf{C}_d^{FE} &= \mathbf{C} \\
\mathbf{D}_d^{FE} &= \mathbf{D}
\end{aligned}$$

2.3.2 Backward Euler

The second variation of Euler's method has the same foundation, but looking backward from kT to $kT - T$. The backward Euler method differs from the forward Euler method in that it is an implicit method, i.e. solving an ODE as a function of both the current state and the state at a later time.

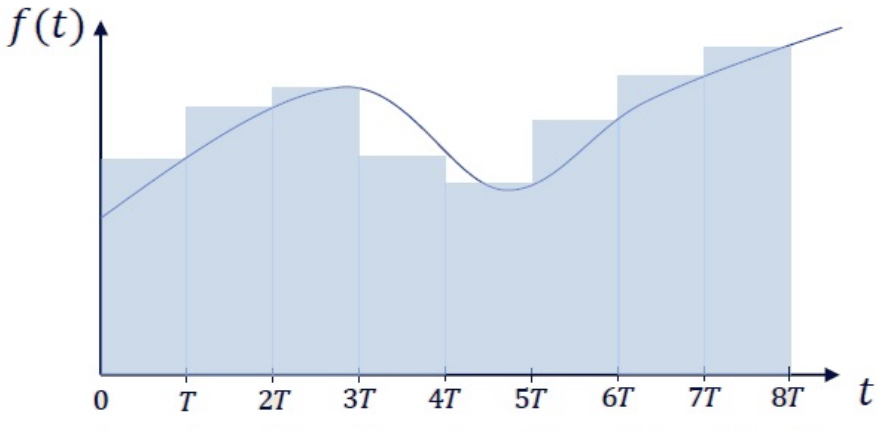


Figure 2.6: Backward Euler approximation of the integral of $f(t)$. Taken from [9]

The backward Euler approximation is derived in a similar fashion to the forward Euler method, however the approximation of the integral in equation (2.19) is made using the right Riemann sum. Applied to a generic ODE this equates to:

$$x((k+1)T) - x(kT) = \int_{kT}^{(k+1)T} f(\tau)\, d\tau \approx T f(x((k+1)T)) \tag{2.23}$$

The backward Euler approximation can be applied to the state-space model in order to discretize the system's equations as it allows for the approximation of $\dot{x}(t)$ in equation

(2.6) in the following manner:

$$\dot{x}((k+1)T) \cong \frac{x((k+1)T) - x(kT)}{T} \tag{2.24}$$

The continuous state-space representation is then transformed into

$$\frac{x((k+1)T) - x(kT)}{T} = \mathbf{A}x((k+1)T) + \mathbf{B}u((k+1)T) \tag{2.25}$$

$$y(kT) = \mathbf{C}x(kT) + \mathbf{D}u(kT) \tag{2.26}$$

Solving for $x(kT)$ and defining a change of variable, $w((k+1)T)$ equation (2.27) is obtained

$$x((k+1)T) - T\mathbf{A}x((k+1)T) - T\mathbf{B}u((k+1)T) = x(kT) \triangleq w((k+1)T) \tag{2.27}$$

Collecting terms and multiplying with the matrix inverse of the factor on either side allows the solving of (2.27) for x in terms of w and u:

$$(\mathbf{I} - T\mathbf{A})x((k+1)T) = w((k+1)T) + T\mathbf{B}u((k+1)T)$$

$$x((k+1)T) = (\mathbf{I} - T\mathbf{A})^{-1}w((k+1)T) + (\mathbf{I} - T\mathbf{A})^{-1}T\mathbf{B}u((k+1)T)$$

Rewriting the above expression at sample kT instead of $(k+1)T$ yields:

$$x(kT) = (\mathbf{I} - T\mathbf{A})^{-1}w(kT) + (\mathbf{I} - T\mathbf{A})^{-1}T\mathbf{B}u(kT) \tag{2.28}$$

which equals the discrete-time state-space equation (2.29) with $w(kT)$ as the new state variable replacing $x(kT)$.

$$w((k+1)T) = (\mathbf{I} - T\mathbf{A})^{-1}w(kT) + (\mathbf{I} - T\mathbf{A})^{-1}T\mathbf{B}u(kT) \tag{2.29}$$

The output equation must also be updated by replacing $x(kT)$ in (2.26) with equation (2.29):

$$y(kT) = \mathbf{C}(\mathbf{I} - T\mathbf{A})^{-1})w(kT) + \mathbf{C}(\mathbf{I} - T\mathbf{A})^{-1}T\mathbf{B}u(kT) + \mathbf{D}u(kT) \tag{2.30}$$

Hence, the backward Euler approximation of the state-space model is given by:

$$w((k+1)T) = \mathbf{A}_d^{BE}w(kT) + \mathbf{B}_d^{BE}u(kT)$$

$$y(kT) = \mathbf{C}_d^{BE}w(kT) + \mathbf{D}_d^{BE}u(kT)$$

where the discrete-time matrices are given by:

$$\mathbf{A}_d^{BE} = (\mathbf{I} - T\mathbf{A})^{-1}$$
$$\mathbf{B}_d^{BE} = (\mathbf{I} - T\mathbf{A})^{-1}T\mathbf{B}$$
$$\mathbf{C}_d^{BE} = \mathbf{C}(\mathbf{I} - T\mathbf{A})^{-1}$$
$$\mathbf{D}_d^{BE} = \mathbf{C}(\mathbf{I} - T\mathbf{A})^{-1}T\mathbf{B} + \mathbf{D}$$

2.3.3 Tustin approximation

Tustin approximation, also known as the bilinear transform is another way of transforming a continuous-time system into a discrete-time representation. The method is based on approximating the integral of a generic ODE, $f(t)$ by the average values of $f(t)$ at the left and right endpoints of the previously considered rectangles, $f(kT)$ and $f((k+1)T)$. See figures 2.5 and 2.6.

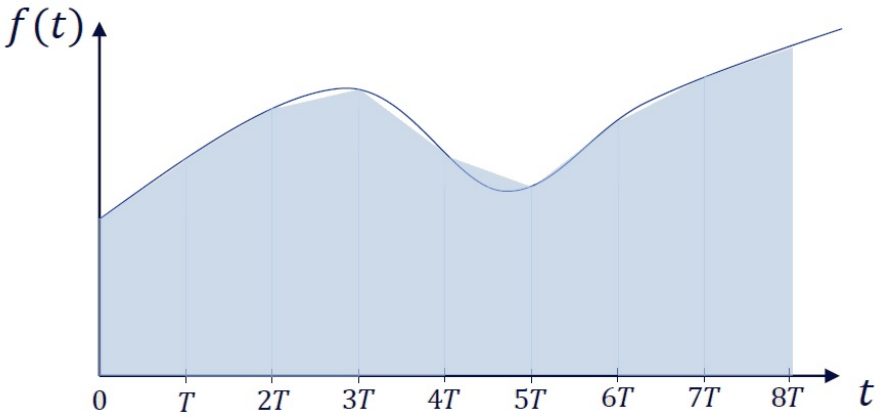


Figure 2.7: Tustin approximation of the integral of $f(t)$. Taken from [10]

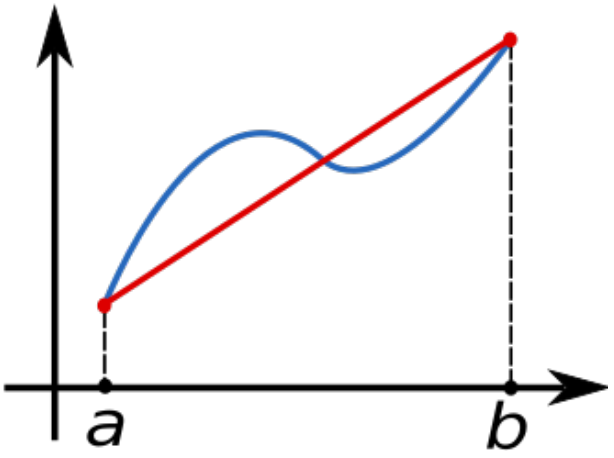


Figure 2.8: The trapezoidal rule works by approximating the region under the graph by as the area of a trapezoid bounded by the endpoints a and b and a linear function (red line) . Image taken from [10]

If the distance between a and b is said to be T units apart the area of a trapezoid, A in Fig.2.8, above is given by the formula:

$$A = \frac{a+b}{2}T$$

Thus, the integral in equation (2.19) can be approximated by:

$$x((k+1)T) - x(kT) = \int_{kT}^{(k+1)T} f(\tau)\, d\tau \cong T\frac{f(x(kT) + f(x((k+1)T)}{2} \tag{2.31}$$

Rearranging equation (2.31) so the expression for $\dot{x}(t)$ is on the left hand side yields:

$$\frac{x((k+1)T) - x(kT)}{T} \cong \frac{f(x(kT) + f(x((k+1)T)}{2} = \frac{\dot{x}(kT) + \dot{x}((k+1)T)}{2} \tag{2.32}$$

The Tustin approximation can then be applied to the state-space model in order to discretize the system's equations by replacing $\dot{x}(t)$ in (2.6) with (2.32) in the following manner:

$$\frac{x((k+1)T) - x(kT)}{T} = \frac{\dot{x}(kT) + \dot{x}((k+1)T)}{2} \tag{2.33}$$

The continuous-time state-space representation is then transformed into:

$$\frac{x((k+1)T) - x(kT)}{T} = \frac{\mathbf{A}(x(kT) + x((k+1)T)) + \mathbf{B}(u(kT) + u((k+1)T))}{2} \tag{2.34}$$

$$y(kT) = \mathbf{C}x(kT) + \mathbf{D}u(kT) \tag{2.35}$$

Rearranging the state equations such that all terms containing $(k+1)T$ on the left side and kT on the right side yields:

$$x((k+1)T) - \frac{T\mathbf{A}}{2}x((k+1)T) - \frac{T\mathbf{B}}{2}u((k+1)T) = x(kT) + \frac{T\mathbf{A}}{2}x(kT) + \frac{T\mathbf{B}}{2}u(kT) \tag{2.36}$$

A change of variable is performed, introducing the scale factor $\sqrt{T}$ to balance the gain of the discrete equivalent between input and output.

$$\sqrt{T}w((k+1)T) \triangleq x(kT) + \frac{T\mathbf{A}}{2}x(kT) + \frac{T\mathbf{B}}{2}u(kT) \tag{2.37}$$

$$x((k+1)T) - \frac{T\mathbf{A}}{2}x((k+1)T) - \frac{T\mathbf{B}}{2}u((k+1)T) = \sqrt{T}w((k+1)T) \tag{2.38}$$

Rewriting the above expression at sample kT instead of $(k+1)T$, and solving for $x(kT)$

$$x(kT) - \frac{T\mathbf{A}}{2}x(kT) - \frac{T\mathbf{B}}{2}u(kT) = \sqrt{T}w(kT) \tag{2.39}$$

$$x(kT) = \left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1}\sqrt{T}w(kT) + \left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1}\frac{T\mathbf{B}}{2}u(kT) \tag{2.40}$$

Combining (2.40) and (2.37):

$$\sqrt{T}w((k+1)T) = \left(\mathbf{I} + \frac{T\mathbf{A}}{2}\right)\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1}\left\{\sqrt{T}w(kT) + \frac{T\mathbf{B}}{2}u(kT)\right\} + \frac{T\mathbf{B}}{2}u(kT) \tag{2.41}$$

$$\Downarrow$$

$$w((k+1)T) = \left(\mathbf{I} + \frac{T\mathbf{A}}{2}\right)\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1}w(kT) + \left[\left(\mathbf{I} + \frac{T\mathbf{A}}{2}\right)\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1} + \mathbf{I}\right]\frac{1}{2}\sqrt{T}\mathbf{B}u(kT) \tag{2.42}$$

The Tustin discretized state-space equations can be found by rewriting equation (2.42) using the following identity:

$$\begin{aligned}\left(\mathbf{I} + \frac{T\mathbf{A}}{2}\right)\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1} + \mathbf{I} &= \left(\mathbf{I} + \frac{T\mathbf{A}}{2}\right)\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1} + \left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1} \\ &= \left(\mathbf{I} + \frac{T\mathbf{A}}{2} + \mathbf{I} - \frac{T\mathbf{A}}{2}\right)\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1} \\ &= 2\mathbf{I}\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1}\end{aligned} \tag{2.43}$$

$$\Downarrow$$

$$w((k+1)T) = \left(\mathbf{I} + \frac{T\mathbf{A}}{2}\right)\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1}w(kT) + \left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1}\sqrt{T}\mathbf{B}u(kT) \tag{2.44}$$

The output equation must also be updated by replacing $x(kT)$ in (2.35) with the equivalent (2.40):

$$y(kT) = \mathbf{C}\left(\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1}\sqrt{T}w(kT) + \left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1}\frac{T\mathbf{B}}{2}u(kT)\right) + \mathbf{D}u(kT)$$

$$\Downarrow$$

$$y(kT) = \mathbf{C}\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1}\sqrt{T}w(kT) + \left[\mathbf{C}\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1}\frac{T\mathbf{B}}{2} + \mathbf{D}\right]u(kT) \tag{2.45}$$

The Tustin approximation of the state-space model is given by (2.44) and (2.45):

$$w((k+1)T) = \mathbf{A}_d^{TU} w(kT) + \mathbf{B}_d^{TU} u(kT)$$

$$y(kT) = \mathbf{C}_d^{TU} w(kT) + \mathbf{D}_d^{TU} u(kT)$$

where the discrete-time matrices are given by:

$$\begin{aligned}
\mathbf{A}_d^{TU} &= \left(\mathbf{I} + \frac{T\mathbf{A}}{2}\right)\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1} \\
\mathbf{B}_d^{TU} &= \left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1} \sqrt{T}\mathbf{B} \\
\mathbf{C}_d^{TU} &= \mathbf{C}\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1} \sqrt{T} \\
\mathbf{D}_d^{TU} &= \mathbf{C}\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1} \frac{T\mathbf{B}}{2} + \mathbf{D}
\end{aligned}$$

2.3.4 Zero order hold

Data hold is a commonly used mathematical model of the signal reconstruction achieved from conventional D/A converters. It is done as a way generating a continuous-time signal, $h(t)$ from a discrete-time signal, $x(kT)$. The conversion works by approximating $h(t)$ in the time interval $kT \leq t < h(k+1)T$ as a polynomial in τ:

$$h(kT+\tau) = a_n\tau^n + a_{n-1}\tau^{n-1} + ... + a_1\tau + a_0$$

Where $0 \leq \tau \leq T$:

$$h(kT) = x(kT)$$

such that:

$$h(kT+\tau) = a_n\tau^n + a_{n-1}\tau^{n-1} + ... + a_1\tau + x(kT)$$

The order of the hold is defined by the order of the polynomial, n. This thesis will only consider the zero-order hold (ZOH) that is obtained when $n = 0$:

$$h(kT+\tau) = x(kT), \quad \tau\epsilon[0,T], \quad k = 0,1,2,...$$

Higher order holds provide increased accuracy in approximating the continuous-time signal at the cost of increasing the time delay and the complexity.Fig.2.9, shows an example of a signal reconstructed by a ZOH.

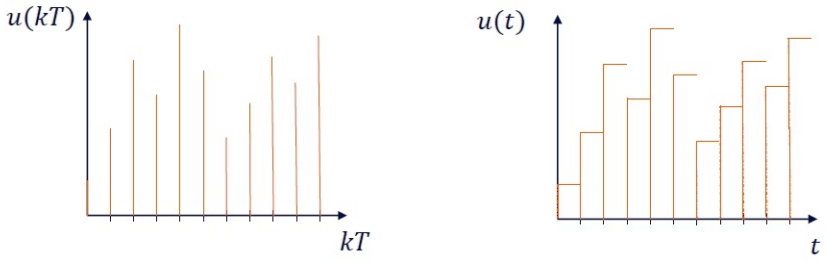


Figure 2.9: By setting $u(t) = u(kT)$ in the interval $kT \leq t < (k+1)T$ a discrete-time signal can be transformed into a continuous-time signal using the ZOH transform. Taken from [9]

The ZOH can also be used as a way of discretizing a continuous-time LTI state-space model:

$$\begin{aligned} \dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t) \end{aligned} \tag{2.46}$$

$$\Downarrow$$

$$\begin{aligned} x((k+1)T) &= \mathbf{A}_d^{ZOH}x(kT) + \mathbf{B}_d^{ZOH}u(kT) \\ y(kT) &= \mathbf{C}_d^{ZOH}x(kT) + \mathbf{D}_d^{ZOH}u(kT) \end{aligned} \tag{2.47}$$

By assuming the input vector $u(t)$ only changes at the equally spaced sampling times, T (ZOH):

$$u(t) = u(kT) \quad \text{for} \quad kT \leq t < (k+1)T \tag{2.48}$$

and expressing the LTI system as the complete solution of (2.46) given by equation (2.49)[1]:

$$x(t) = e^{\mathbf{A}t}x(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)d\tau = e^{\mathbf{A}t}\left(x(0) + \int_0^t e^{-\mathbf{A}\tau}\mathbf{B}u(\tau)d\tau\right) \tag{2.49}$$

[1]The explicit expression for continuous-time LTI system is taken from [11]

$x(kT)$ and $x((k+1)T)$ can be rewritten as:

$$\begin{aligned} x(kT) &= e^{\mathbf{A}kT}\left(x(0) + \int_0^{kT} e^{-\mathbf{A}\tau}\mathbf{B}u(\tau)d\tau\right) \\ x((k+1)T) &= e^{\mathbf{A}(k+1)T}\left(x(0) + \int_0^{(k+1)T} e^{-\mathbf{A}\tau}\mathbf{B}u(\tau)d\tau\right) \\ &= e^{\mathbf{A}T}e^{\mathbf{A}kT}\left(x(0) + \int_0^{kT} e^{-\mathbf{A}\tau}\mathbf{B}u(\tau)d\tau + \int_{kT}^{(k+1)T} e^{-\mathbf{A}\tau}\mathbf{B}u(\tau)d\tau\right) \\ x((k+1)T) &= e^{\mathbf{A}T}x(kT) + e^{\mathbf{A}(k+1)T}\int_{kT}^{(k+1)T} e^{-\mathbf{A}\tau}\mathbf{B}u(\tau)d\tau \\ &= e^{\mathbf{A}T}x(kT) + \int_{kT}^{(k+1)T} e^{\mathbf{A}[(k+1)T-\tau]}\mathbf{B}u(\tau)d\tau \end{aligned} \tag{2.50}$$

Finally the discrete-time matrices that yield the exact values at $t = kT$ can be found by looking at the expression for $x((k+1)T$ obtained after introducing the change of variable $\lambda = (k+1)T - \tau$ into equation (2.50) and taking into account the ZOH assumption (2.48):

$$x((k+1)T) = e^{\mathbf{A}T}x(kT) + \int_0^T e^{\mathbf{A}\lambda}\mathbf{B}u(kT)d\lambda \tag{2.51}$$

The discrete-time version of the output and feedthrough matrices from (2.46) remain unchanged. Thus, the discretized model (2.47) is given by the matrices:

$$\begin{aligned} \mathbf{A}_d^{ZOH} &= e^{\mathbf{A}T} \\ \mathbf{B}_d^{ZOH} &= \int_0^T e^{\mathbf{A}\lambda}\mathbf{B}d\lambda \\ \mathbf{C}_d^{ZOH} &= \mathbf{C} \\ \mathbf{D}_d^{ZOH} &= \mathbf{D} \end{aligned}$$

Unlike the discretization techniques discussed previously in sections 2.3.1, 2.3.2 and 2.3.3 the discrete representation (2.51) contains no approximation, meaning it calculates the exact values of the state vector at times $t = kT$ thus, the method perfectly represents the evolution of the state vector given the assumption that $u(t)$ is constant between consecutive time samples (2.48) holds true.

2.3.5 Investigating a special case for the hover model

When comparing the results of the different discretized models special case is observed. When calculating the discrete-time matrices of the hover model the Tustin approximation and the Zero-order hold method yield the same $\mathbf{A}$ and $\mathbf{B}$ matrices:

$$\begin{aligned} \mathbf{A}_d^{ZOH} = e^{\mathbf{A}T} &= \mathbf{A}_d^{TU} = \left(\mathbf{I} + \frac{T\mathbf{A}}{2}\right)\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1} \\ \mathbf{B}_d^{ZOH} = \int_0^T e^{\mathbf{A}\lambda}\mathbf{B}d\lambda &= \mathbf{B}_d^{TU} = \left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1}\sqrt{T}\mathbf{B} \end{aligned}$$

In section 2.2 the state matrix for the hover was found to be the non diagonalizable matrix, $\mathbf{A}$:

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

The discrete-time state matrix $\mathbf{A}_d^{TU}$ is obtained by solving:

$$\left(\mathbf{I} + \frac{T\mathbf{A}}{2}\right) = \begin{bmatrix} 1 & 0 & 0 & T/2 & 0 & 0 \\ 0 & 1 & 0 & 0 & T/2 & 0 \\ 0 & 0 & 1 & 0 & 0 & T/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1}$$

$$\Downarrow$$

$$\left(\mathbf{I} + \frac{T\mathbf{A}}{2}\right)\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The matrix exponential of a given matrix $\mathbf{A}T$ is found by solving the power series:

$$e^{\mathbf{A}T} = \sum_{k=0}^{\infty} \frac{1}{k!}(\mathbf{A}T)^k \tag{2.52}$$

A nilpotent matrix is a nth order square matrix, N, that when raised to some integer k referred to as its index results in the zero matrix. More generally any nth order triangular matrix with zeros along its main diagonal is nilpotent with index, $k \leq n$. The hover state matrix $\mathbf{A}$ is nilpotent with index 2 and therefore:

$$A^j = 0 \quad \forall \quad j \geq k$$

$$\mathbf{A^2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Hence, equation (2.52) turns into an expression containing only two non-zero terms:

$$\sum_{k=0}^{\infty} \frac{1}{k!}(\mathbf{A}T)^k = \frac{1}{0!}(\mathbf{A}T)^0 + \frac{1}{1!}(\mathbf{A}T)^1 + \frac{1}{2!} * 0 + 0 + ... = \mathbf{I} + \mathbf{A}T \tag{2.53}$$

$$\Downarrow$$

$$\mathbf{I} + \mathbf{A}T = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For the input matrix $\mathbf{B}$:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{LK_f}{J_p} & -\frac{LK_f}{J_p} & 0 & 0 \\ 0 & 0 & -\frac{LK_f}{J_r} & \frac{LK_f}{J_r} \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} \end{bmatrix}$$

the discrete-time matrix $\mathbf{A}_d^{TU}$ is found by solving the equation:

$$\left(\mathbf{I} - \frac{T\mathbf{A}}{2}\right)^{-1} \sqrt{T}\mathbf{B}$$

$$\mathbf{B}_d^{TU} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \sqrt{T} * \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{LK_f}{J_p} & -\frac{LK_f}{J_p} & 0 & 0 \\ 0 & 0 & -\frac{LK_f}{J_r} & \frac{LK_f}{J_r} \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} \end{bmatrix} \tag{2.54}$$

$$\Downarrow$$

$$\mathbf{B}_d^{TU} = \begin{bmatrix} -\frac{K_t T\sqrt{T}}{2J_y} & -\frac{K_t T\sqrt{T}}{2J_y} & \frac{K_t T\sqrt{T}}{2J_y} & \frac{K_t T\sqrt{T}}{2J_y} \\ \frac{K_f LT\sqrt{T}}{2J_p} & -\frac{K_f LT\sqrt{T}}{2J_p} & 0 & 0 \\ 0 & 0 & \frac{K_f LT\sqrt{T}}{2J_r} & -\frac{K_f LT\sqrt{T}}{2J_r} \\ -\frac{K_t\sqrt{T}}{J_y} & -\frac{K_t\sqrt{T}}{J_y} & \frac{K_t\sqrt{T}}{J_y} & \frac{K_t\sqrt{T}}{J_y} \\ \frac{K_f L\sqrt{T}}{J_p} & -\frac{K_f L\sqrt{T}}{J_p} & 0 & 0 \\ 0 & 0 & \frac{K_f L\sqrt{T}}{J_r} & -\frac{K_f L\sqrt{T}}{J_r} \end{bmatrix} \tag{2.55}$$

Using Matlab's symbolic math toolbox to calculate $\mathbf{B}_d^{ZOH}$ yields the following result:

$$\mathbf{B}_d^{ZOH} = \begin{bmatrix} -\frac{K_t T^2}{2J_y} & -\frac{K_t T^2}{2J_y} & \frac{K_t T^2}{2J_y} & \frac{K_t T^2}{2J_y} \\ \frac{K_f LT^2}{2J_p} & -\frac{K_f LT^2}{2J_p} & 0 & 0 \\ 0 & 0 & \frac{K_f LT^2}{2J_r} & -\frac{K_f LT^2}{2J_r} \\ -\frac{K_t T}{J_y} & -\frac{K_t T}{J_y} & \frac{K_t T}{J_y} & \frac{K_t T}{J_y} \\ \frac{K_f LT}{J_p} & -\frac{K_f LT}{J_p} & 0 & 0 \\ 0 & 0 & \frac{K_f LT}{J_r} & -\frac{K_f LT}{J_r} \end{bmatrix} \tag{2.56}$$

The matrices appear different at first, however state-space representations are not unique. In can be shown that two state-space models, $sys1(A_1, B_1, C_1, D_1)$ and $sys2(A_2, B_2, C_2, D_2)$ are representations of the same system if there exist a transformation matrix $\mathbf{T_f}$ such that:

$$\mathbf{A_1} = \mathbf{T_f}\mathbf{A_2}{\mathbf{T_f}}^{-1} \tag{2.57}$$

$$\mathbf{B_1} = \mathbf{T_f}\mathbf{B_2} \tag{2.58}$$

$$\mathbf{C_1} = \mathbf{C_2}{\mathbf{T_f}}^{-1} \tag{2.59}$$

$$\mathbf{D_1} = \mathbf{D_2} \tag{2.60}$$

A closer look at B_d^{TU} and B_d^{ZOH} reveals that there exists transformation matrix $\mathbf{T_f} = \mathbf{I}\sqrt{T}$ where T is the periodic sampling interval. This matrix satisfies (2.57) and (2.58) The systems are in fact not identical as seen in the comparisons made in 3, only the state and input matrices which are used to calculate the control gain matrix $\mathbf{K}$ (4.6) are. Resulting in equal performance as calculated by the cost function, J given by equation (4.3).

Chapter 3

Comparison of the discretized models

This section will analyse the different discretized models derived in section 2.3. At small values of T the models are essentially equivalent, this is shown in Figure 3.1 where $T = 0.02s$. For perspective CT functions in MATLAB has a sampling time of $T = 0.002s$. Therefore the models will be compared to each other based on stability properties and how well they fit the original model as T increases and the differences become more noticeable as seen in Figure 3.2. As T increases performance may decline and in some cases instability can be introduced. The figures presented in this section is made by discretizing the CT system (2.11) with arbitrary inputs, $u(k)$.

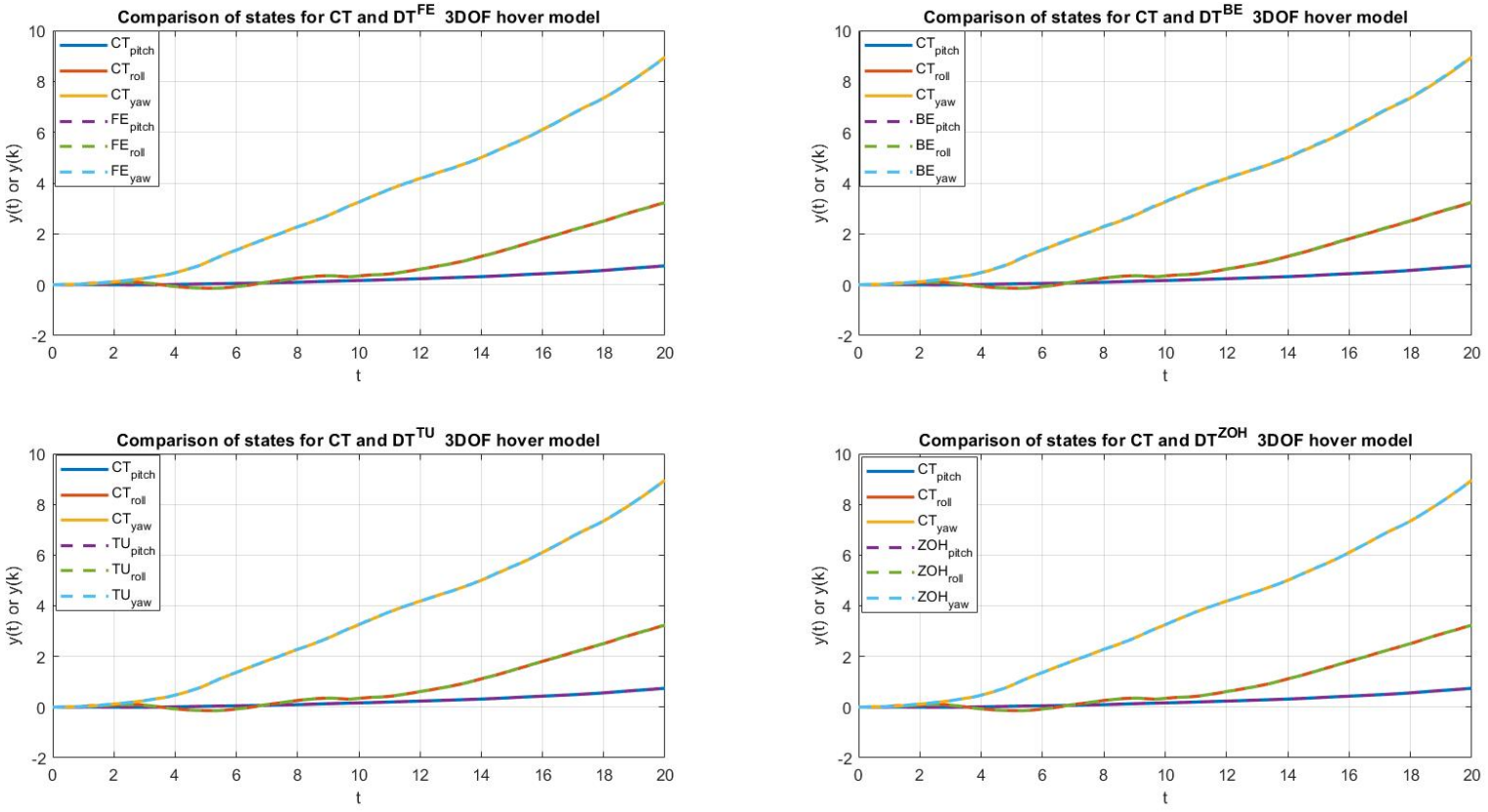


Figure 3.1: The CT and DT models simulated with arbitrary inputs, $u(k)$ and $u(t)$ are indistinguishable for a small sampling time of $T = 0.02s$.

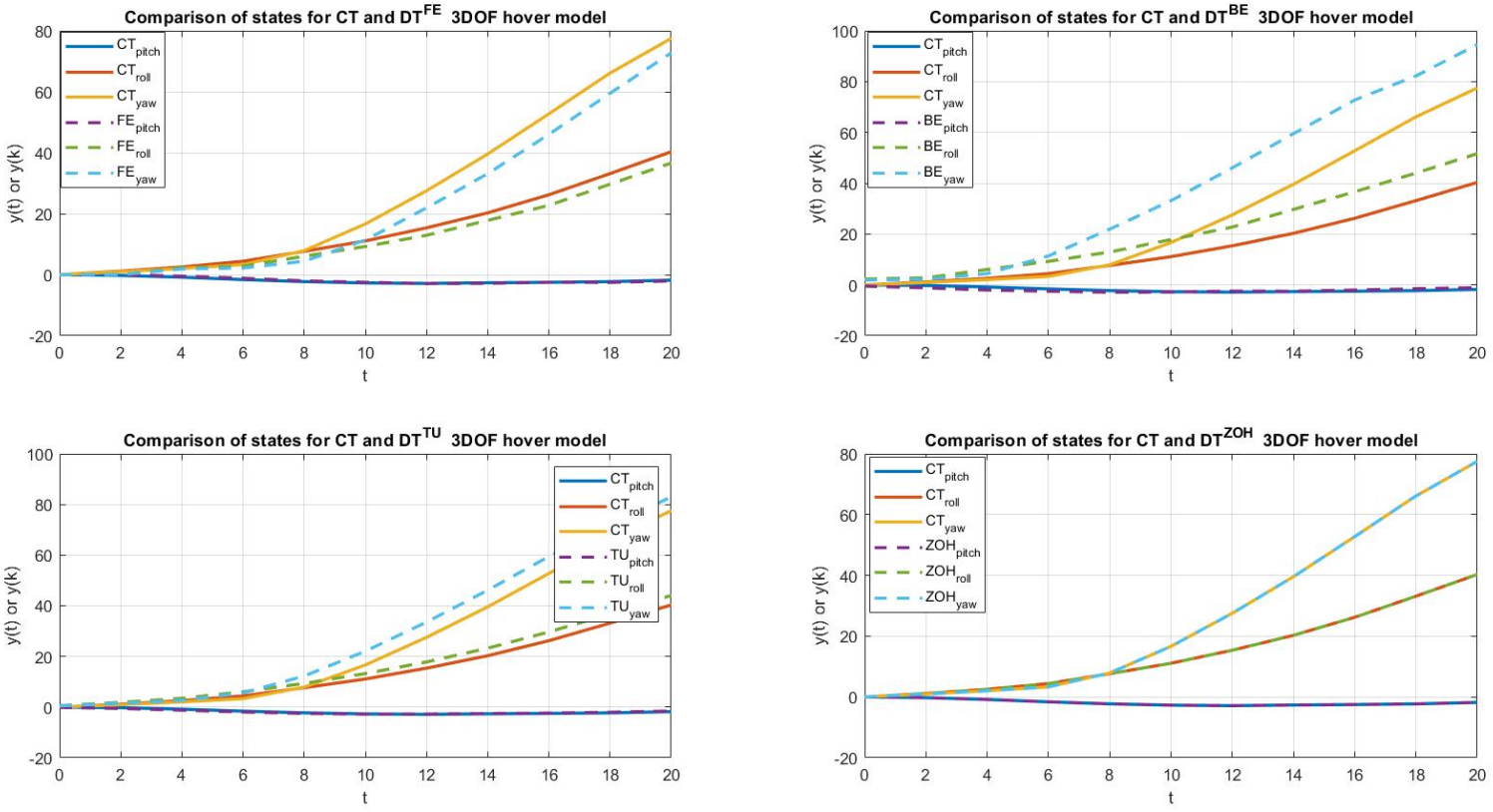


Figure 3.2: The CT and DT models simulated with the same inputs, $u(k)$ and $u(t)$ as in 3.1 with $T = 2s$. The deviations from the CT system are noticeable in all cases except for ZOH.

3.1 Stability

When considering the stability of a system based on the eigenvalues the criteria differ for CT and DT systems.

Consider the open loop system $\dot{x} = Ax \in \mathbb{R}^n$. For a CT LTI system it is simple to solve for $x(t)$ at some later point based on initial eigenvalues and an initial condition $x(0)$:

$$x(t) = \mathbf{V}e^{\mathbf{D}t}\mathbf{V}^{-1}x(0) \tag{3.1}$$

Where $\mathbf{D} \in \mathbb{C}^{n\times n}$ is a matrix with eigenvalues, λ_i along its diagonal and $\mathbf{V}$ is a matrix containing the corresponding right eigenvectors such that $\mathbf{AV} = \mathbf{VD}$ where column i in $\mathbf{V} = v_i$. This solves the eigenvalue problem:

$$\mathbf{A}v_i = \lambda_i v_i \tag{3.2}$$

$$\mathbf{D} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} \tag{3.3}$$

Looking at equation (3.1) it can be shown that the stability of the system is decided by the matrix exponential, $e^{\mathbf{D}t}$.

$$e^{\mathbf{D}t} = \begin{bmatrix} e^{\lambda_1 t} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2 t} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & e^{\lambda_n t} \end{bmatrix} \tag{3.4}$$

$$\lambda = a \pm ib \text{ and } e^{\pm\lambda}t = e^{at}\left[cos(bt) \pm isin(bt)\right] \tag{3.5}$$

From (3.5) it is clear that $e^{\mathbf{D}t}$ converges to infinity when the real part of the eigenvalues are $a > 0$ and to zero for $a < 0$ as illustrated in Figure 3.3:

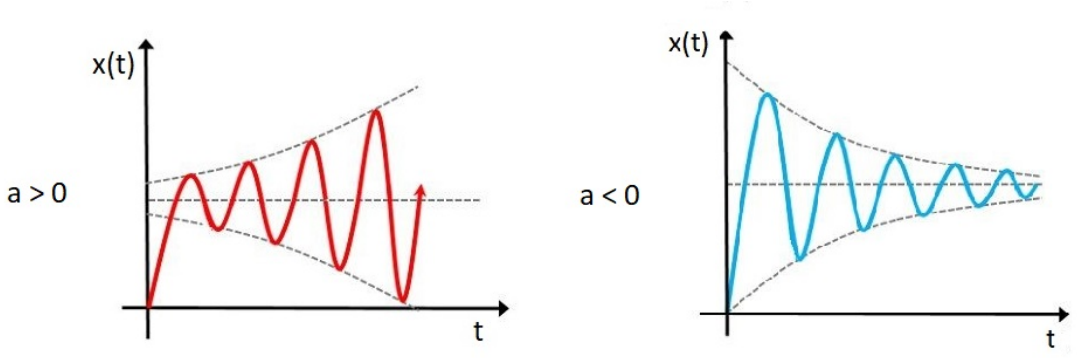


Figure 3.3: Modified illustration showing the effects of the real part of eigenvalues for a CT LTI system, taken from [12]. The red graph is the asymptotically unstable system where $e^{\mathbf{D}t}$ converges to infinity and the blue graph is the asymptotically stable system where the matrix exponential converges to zero.

Given the eigenvalues of $\mathbf{A}$ a question is raised regarding whether the stability properties of the CT system are preserved during discretization or not. In the case of FE discretization where $\mathbf{A}_d^{FE} = \mathbf{I} + \mathbf{TA}$, the DT eigenvalues, λ_i^{FE} can be derived by transforming (3.2) into:

$$\mathbf{A}_d^{FE} v_i = \lambda_i^{FE} v_i \tag{3.6}$$

$$\begin{gathered}
\mathbf{A}v_i = \lambda_i v_i \\
\Downarrow \\
\mathbf{TA}v_i = \mathbf{T}\lambda_i v_i \\
\Downarrow \\
v_i + \mathbf{TA}v_i = v_i + \mathbf{T}\lambda_i v_i \\
\Downarrow \\
\underbrace{(\mathbf{I} + \mathbf{TA})}_{\mathbf{A}_d^{FE}} v_i = \underbrace{(1 + \mathbf{T}\lambda_i)}_{\lambda_i^{FE}} v_i
\end{gathered}$$

Comparing the asymptotic stability region of the CT system $Re(\lambda_i) < 0$ with $Re(\lambda_i^{FE})$ we get the asymptotic stability region $Re(\lambda_i^{FE}) < 1$:

$$\begin{gathered}
1 - TRe(\lambda_i) = Re(\lambda_i^{FE}) \\
\Downarrow \\
Re(\lambda_i) = \frac{1}{T}(Re(\lambda_i^{FE}) - 1) \\
\Downarrow \\
\frac{1}{T}(Re(\lambda_i^{FE}) - 1) < 0 \\
\Downarrow \\
Re(\lambda_i^{FE}) < 1
\end{gathered}$$

As λ_i is mapped onto $\lambda_i^{FE} = 1 + \mathbf{T}\lambda_i$ the CT stability region bound by $Re(\lambda_i) > 0$ is transformed into the region $Re(\lambda_i^{FE}$. Both regions are presented in Figure 3.4. It can be seen that not all parts of the CT stability region is mapped onto the DT equivalent, a

consequence of this is that there is a possibility for a stable CT system could be mapped into an unstable DT system.

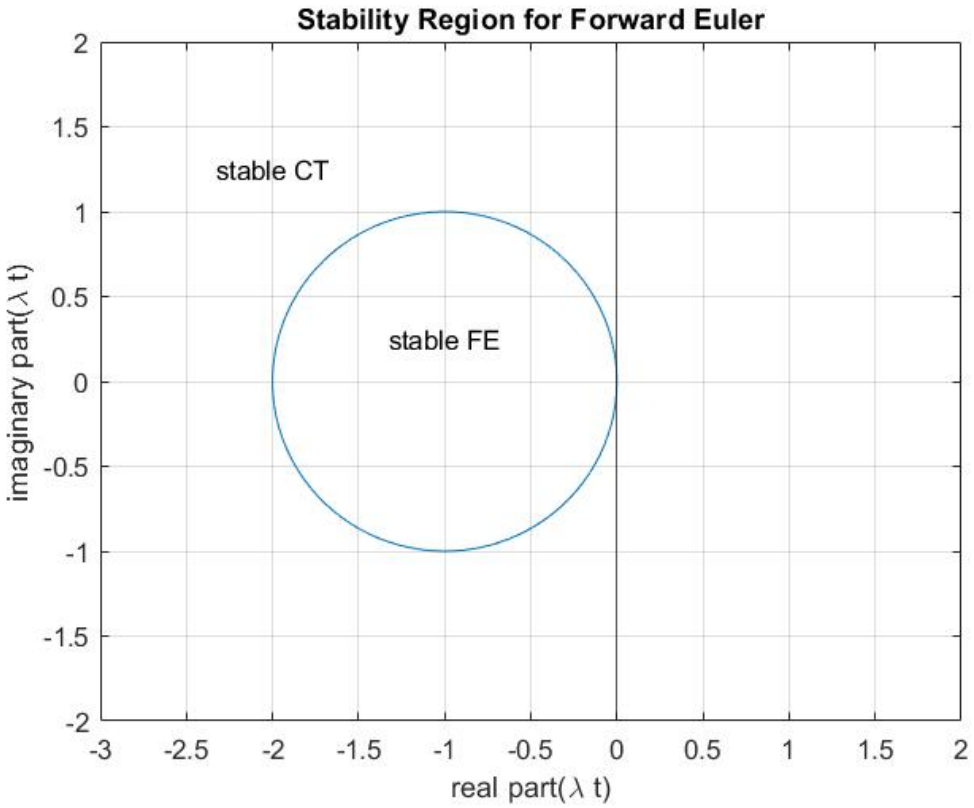


Figure 3.4: Plot of the stability regions of the CT system and the FE discretized model of it.

The eigenvalues of $\mathbf{A}^{BE} = (\mathbf{I} - \mathbf{TA})^{-1}$ are found in similar fashion by transforming $\mathbf{A}$ in (3.1) into and equivalent based on $\mathbf{A}^{BE}$:

$$\mathbf{A}v_i = \lambda_i v_i$$

$$\Downarrow$$

$$-\mathbf{TA}v_i = -\mathbf{T}\lambda_i v_i$$

$$\Downarrow$$

$$v_i - \mathbf{TA}v_i = v_i - \mathbf{T}\lambda_i v_i$$

$$\Downarrow$$

$$(\mathbf{I} - \mathbf{TA})v_i = (1 - \mathbf{T}\lambda_i)v_i$$

$$\Downarrow$$

$$(\mathbf{I}-\mathbf{TA})^{-1}(\mathbf{I}-\mathbf{TA})v_i = (1-\mathbf{T}\lambda_i)(\mathbf{I}-\mathbf{TA})^{-1}v_i$$

$$\Downarrow$$

$$\underbrace{\frac{1}{1-\mathbf{T}\lambda_i}}_{\lambda_i^{BE}} v_i = \underbrace{(\mathbf{I}-\mathbf{TA})^{-1}}_{\mathbf{A}_d^{BE}} v_i$$

Since λ_i is mapped onto $\lambda_i^{BE} = \frac{1}{1-\mathbf{T}\lambda_\mathbf{i}}$ it can be shown that:

$$\lambda_i^{BE} = \frac{1}{1-\mathbf{T}\lambda_i} = \frac{1}{2} + \{\frac{1}{1-\mathbf{T}\lambda_i} - \frac{1}{2}\} = \frac{1}{2} - \frac{1}{2}\frac{1+\mathbf{T}\lambda_i}{1-\mathbf{T}\lambda_i}$$

By substituting λ_i with $jw,\ w \in \mathbb{R}$ the above expression turns into:

$$\lambda_i^{BE} = \frac{1}{2} - \frac{1}{2}\frac{1+\mathbf{T}jw}{1-\mathbf{T}jw}, \quad \text{where } \left|\frac{1+\mathbf{T}jw}{1-\mathbf{T}jw}\right| = 1 \,\forall\, w \in \mathbb{R} \tag{3.7}$$

The stable region in CT is mapped into a circle with radius 0.5 and centered in (0.5,0) as seen in Figure 3.5. With this method any stable CT system will preserve stability in DT, although in some cases unstable CT systems can be mapped into a stable DT system.

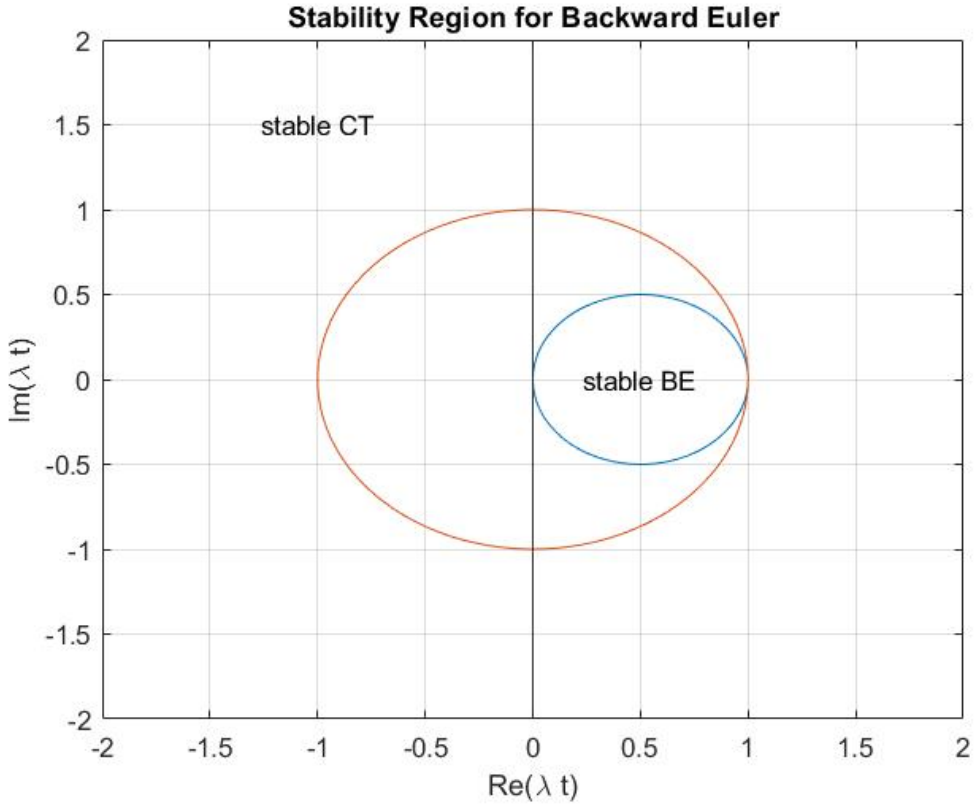


Figure 3.5: Plot of the stability regions of the CT system and the BE discretized model of it.

For the Tustin approximation the eigenvalues of $\mathbf{A}_d^{TU} = (\mathbf{I} + \frac{\mathbf{TA}}{2})(\mathbf{I} - \frac{\mathbf{TA}}{2})^{-1}$ can be found by taking:

$$\mathbf{A}v_i = \lambda_i v_i$$

$$\Downarrow$$

$$\frac{\mathbf{TA}}{2}v_i = \frac{\mathbf{T}\lambda_i}{2}v_i$$

$$\Downarrow$$

$$(\mathbf{I} + \frac{\mathbf{TA}}{2})v_i = (1 + \frac{\mathbf{T}\lambda_i}{2})v_i$$

In combination with:

$$\mathbf{A}v_i = \lambda_i v_i$$

$$\Downarrow$$

$$-\frac{\mathbf{TA}}{2}v_i = -\frac{\mathbf{T}\lambda_i}{2}v_i$$

$$\Downarrow$$

$$(\mathbf{I} - \frac{\mathbf{TA}}{2})v_i = (1 - \frac{\mathbf{T}\lambda_i}{2})v_i$$

$$\Downarrow$$

$$(\mathbf{I} - \frac{\mathbf{TA}}{2})^{-1}v_i = \frac{1}{1 - \frac{\mathbf{T}\lambda_i}{2}}v_i$$

Finally resulting in:

$$(\mathbf{I} + \frac{\mathbf{TA}}{2})(\mathbf{I} - \frac{\mathbf{TA}}{2})^{-1}v_i = (\mathbf{I} + \frac{\mathbf{TA}}{2})\frac{1}{\mathbf{I} - \frac{\mathbf{T}\lambda_\mathbf{i}}{2}}v_i$$

$$\frac{1}{\mathbf{I} - \frac{\mathbf{T}\lambda_\mathbf{i}}{2}}(\mathbf{I} + \frac{\mathbf{TA}}{2})v_i = \frac{(1 + \frac{\mathbf{T}\lambda_i}{2})}{(1 - \frac{\mathbf{T}\lambda_i}{2})}v_i$$

$$\Downarrow$$

$$\lambda_i^{TU} = \frac{(1 + \frac{\mathbf{T}\lambda_i}{2})}{(1 - \frac{\mathbf{T}\lambda_i}{2})}$$

By substituting λ_i with jw, $w \in \mathbb{R}$ the above expression turns into:

$$\lambda_i^{TU} = \frac{1 + \mathbf{T}\lambda_i/2}{1 - \mathbf{T}\lambda_i/2} = \frac{1 + a\mathbf{T}/2 + jw\mathbf{T}/2}{1 - a\mathbf{T}/2 - jw\mathbf{T}/2}$$

Where a is the real part of λ_i and the mapping of the stability region of λ_i^{TU} turns out to be the inside of the unit circle as given by:

$$|\lambda_i^{TU}| = |\frac{1 + a\mathbf{T}/2 + jw\mathbf{T}/2}{1 - a\mathbf{T}/2 - jw\mathbf{T}/2}| = \frac{\sqrt{(1 + a\mathbf{T}/2)^2 + w^2\mathbf{T}^2/4}}{\sqrt{(1 - a\mathbf{T}/2)^2 + w^2\mathbf{T}^2/4}} < 1$$

Squaring both sides of the inequality yields:

$$1 + \frac{a^2T^2}{4} + aT + \frac{w^2T^2}{4} < 1 + \frac{a^2T^2}{4} - aT + \frac{w^2T^2}{4}$$

$$\Downarrow$$

$$2aT < 0 \quad \longrightarrow \quad a < 0$$

This establishes a direct correspondence between the left half-plane of the complex plane for the CT system and the inside of the unit circle for the DT system as seen in Figure 3.6. The stable region in CT is mapped onto the stable region in DT and stability is guaranteed.

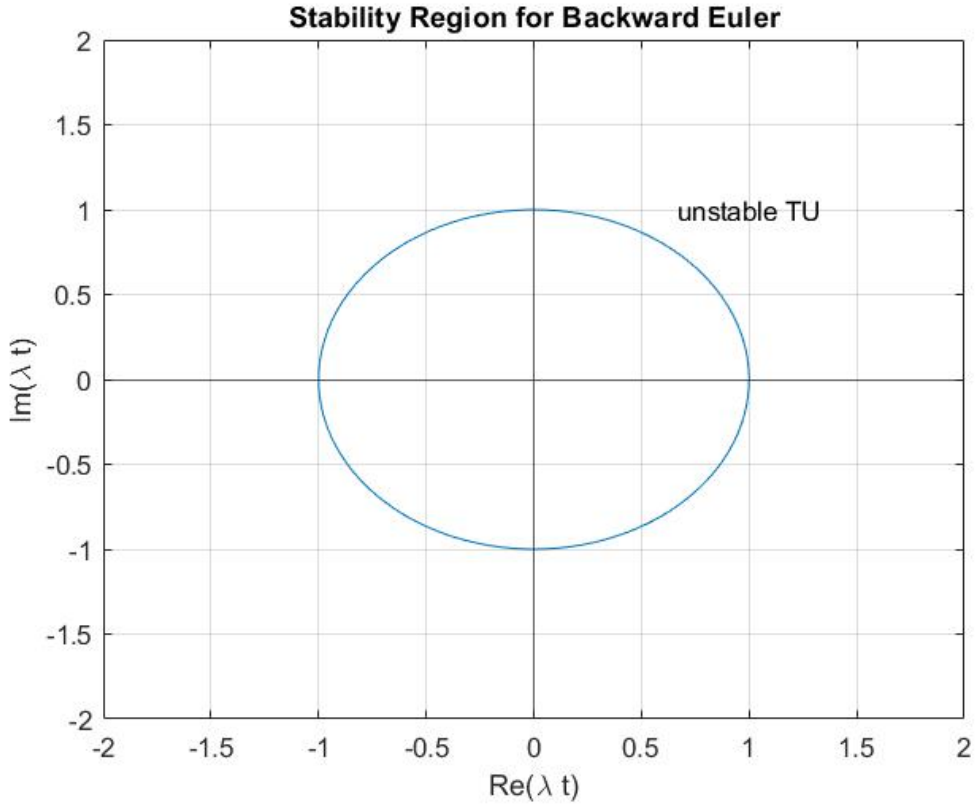


Figure 3.6: Plot of the stability regions of the CT system and the TU discretized model of it.

Finally for the zero-order hold discretization method where $A_d^{ZOH} = e^{\mathbf{A}T}$ the eigendecomposition of $\mathbf{A}$ is given by:

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$$

$$\Downarrow$$

$$A_d^{ZOH} = e^{\mathbf{A}T} = \mathbf{V}e^{\mathbf{D}T}\mathbf{V}^{-1}$$

Where $\mathbf{D}$ is the diagonal matrix with the eigenvalues of $\mathbf{A}$ along its diagonal. This gives the eigenvalues $\lambda_i^{ZOH} = e^{\lambda_i T}$ Mapping the eigenvalues yields:

$$|\lambda_i^{ZOH}| = e^{aT}\sqrt{cos^2 wT + sin^2 wT} = e^{aT} < 1$$

The obtained region is then:

$$a < ln1 = 0 \longrightarrow a < 0$$

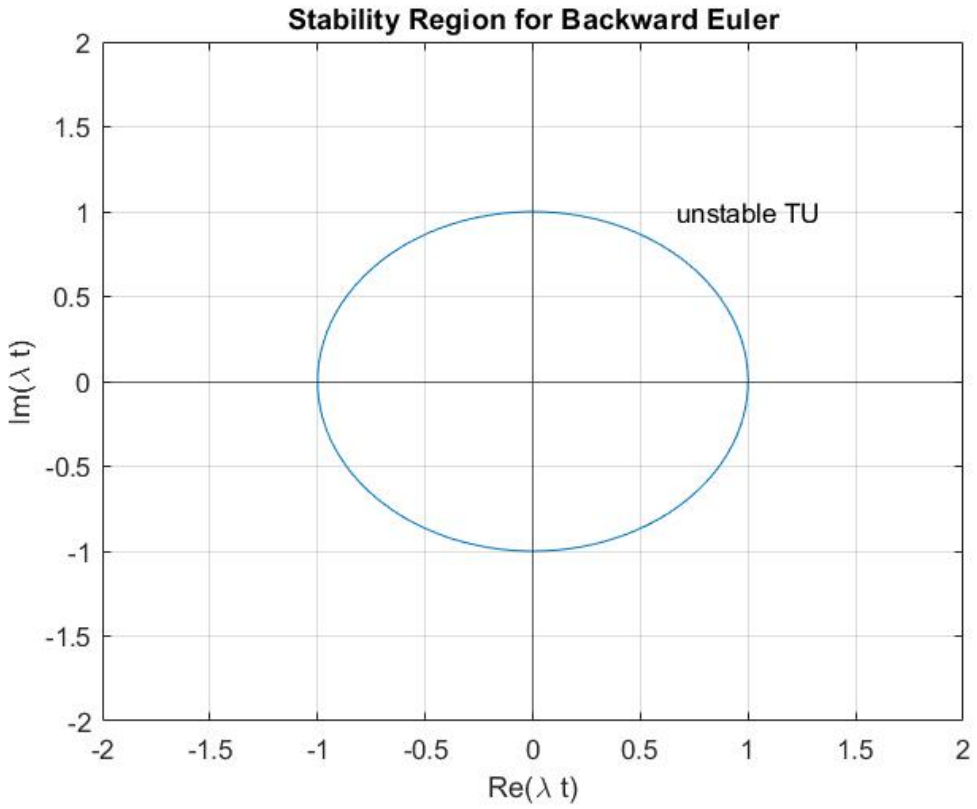


Figure 3.7: Plot of the stability regions of the CT system and the ZOH discretized model of it.

In conclusion the forward Euler discretization technique is the worst method since it does not preserve stability and has lower performance in most cases then the other techniques reviewed. The backwards Euler performes better however it does not compare very well to the Tustin approximation which in turn can not compete against the zero-order hold. This was expected as the ZOH model is a mathematical model of the praxtical signal reconstruction done by conventional digital to analog converter.

3.1.1 Simulated and applied results

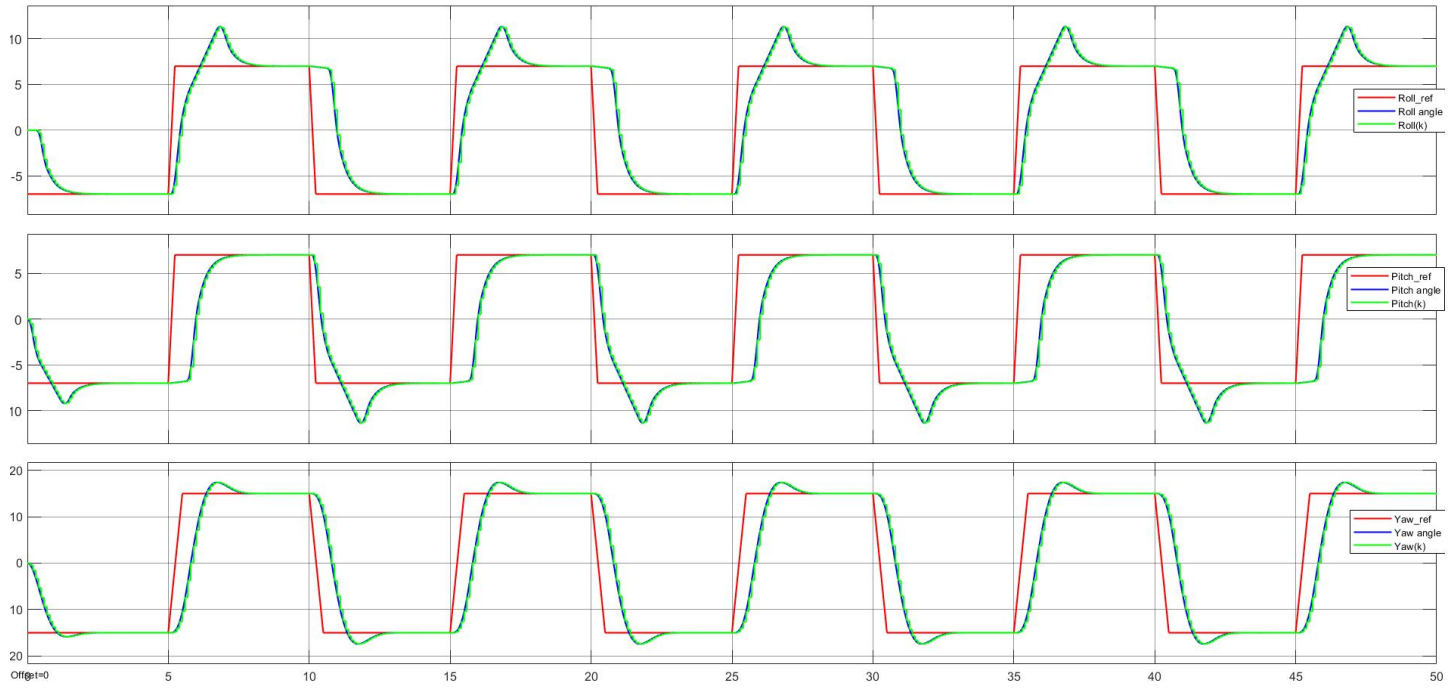


Figure 3.8: Simulated response of FE discretized system with T=0.1.

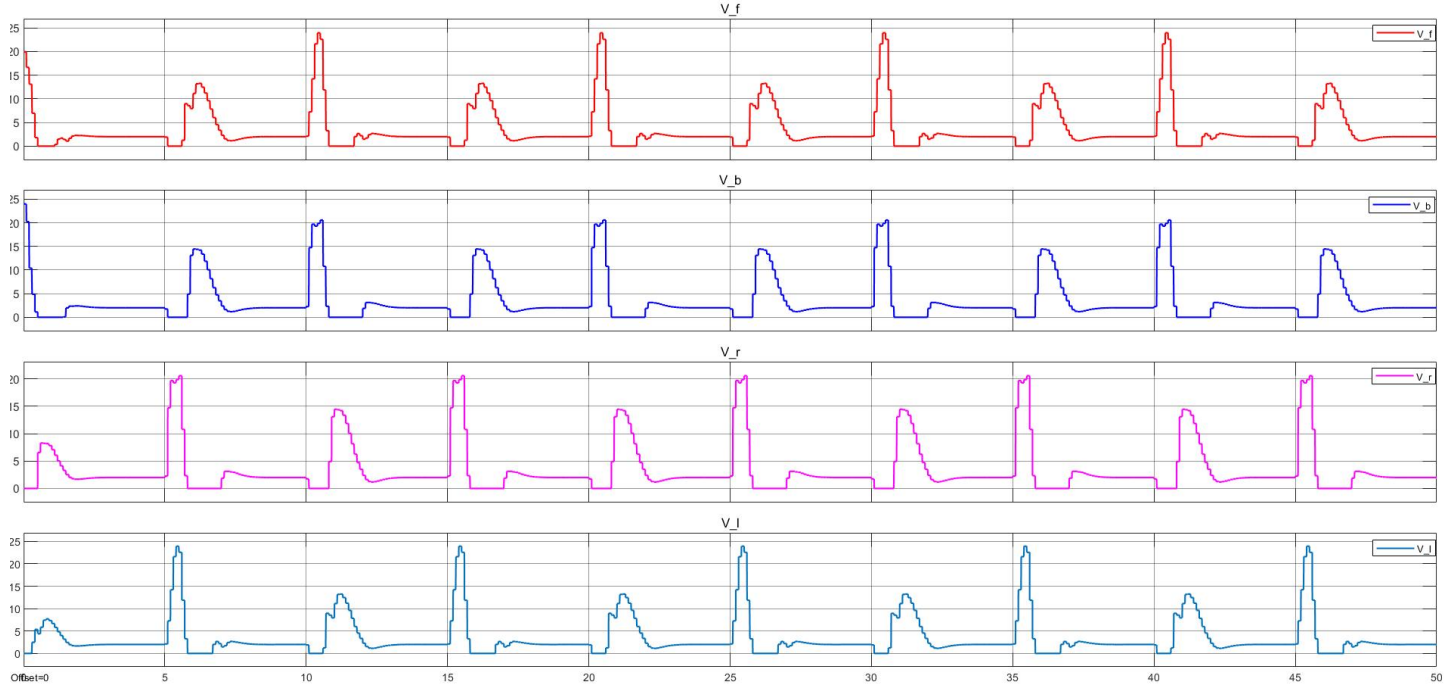


Figure 3.9: Simulated response of FE discretized system with T=0.1.

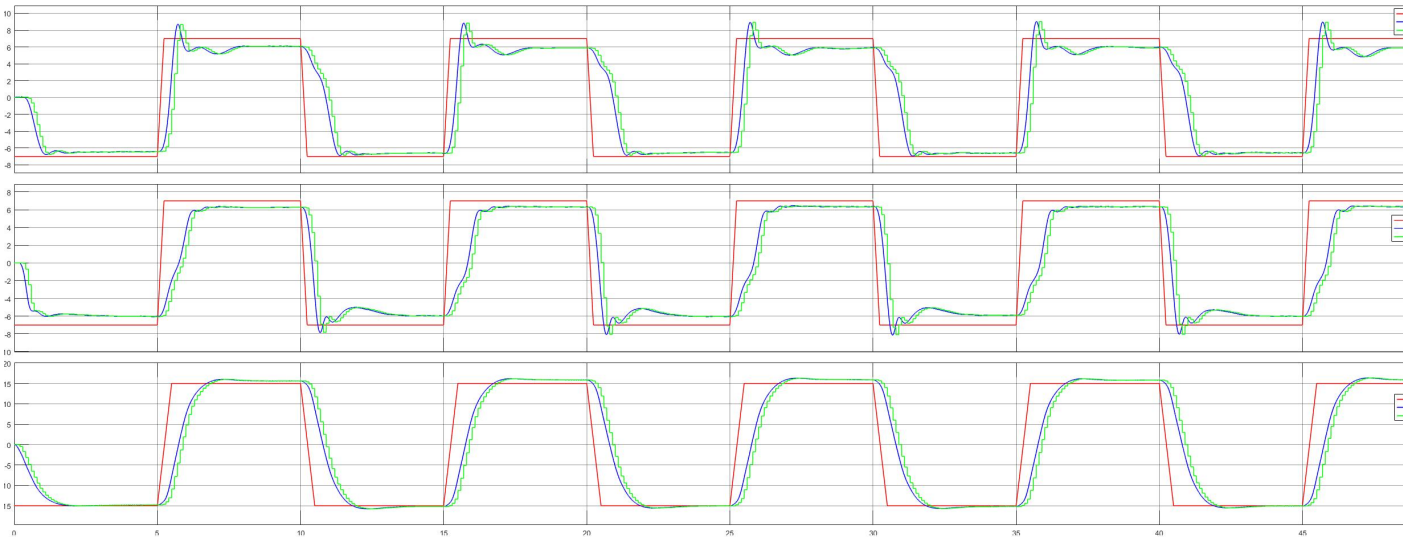


Figure 3.10: Actual response of FE discretized system with T=0.1.

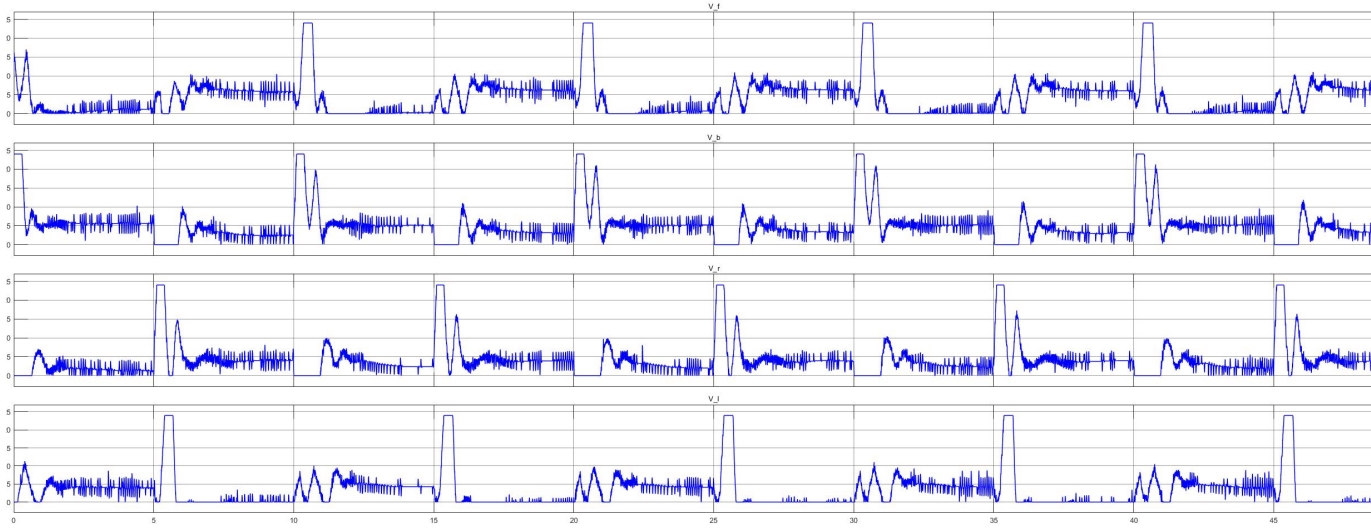


Figure 3.11: Actual response of FE discretized system with T=0.1.

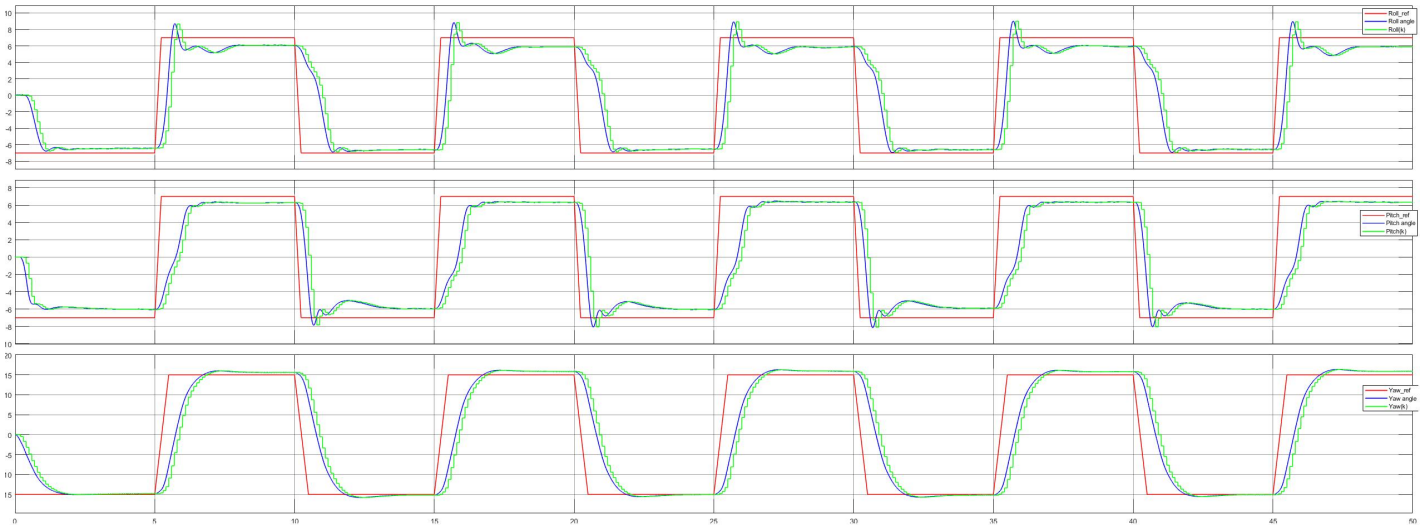


Figure 3.12: Simulated response of event-triggered control when initialized at an elevated pitch angle.

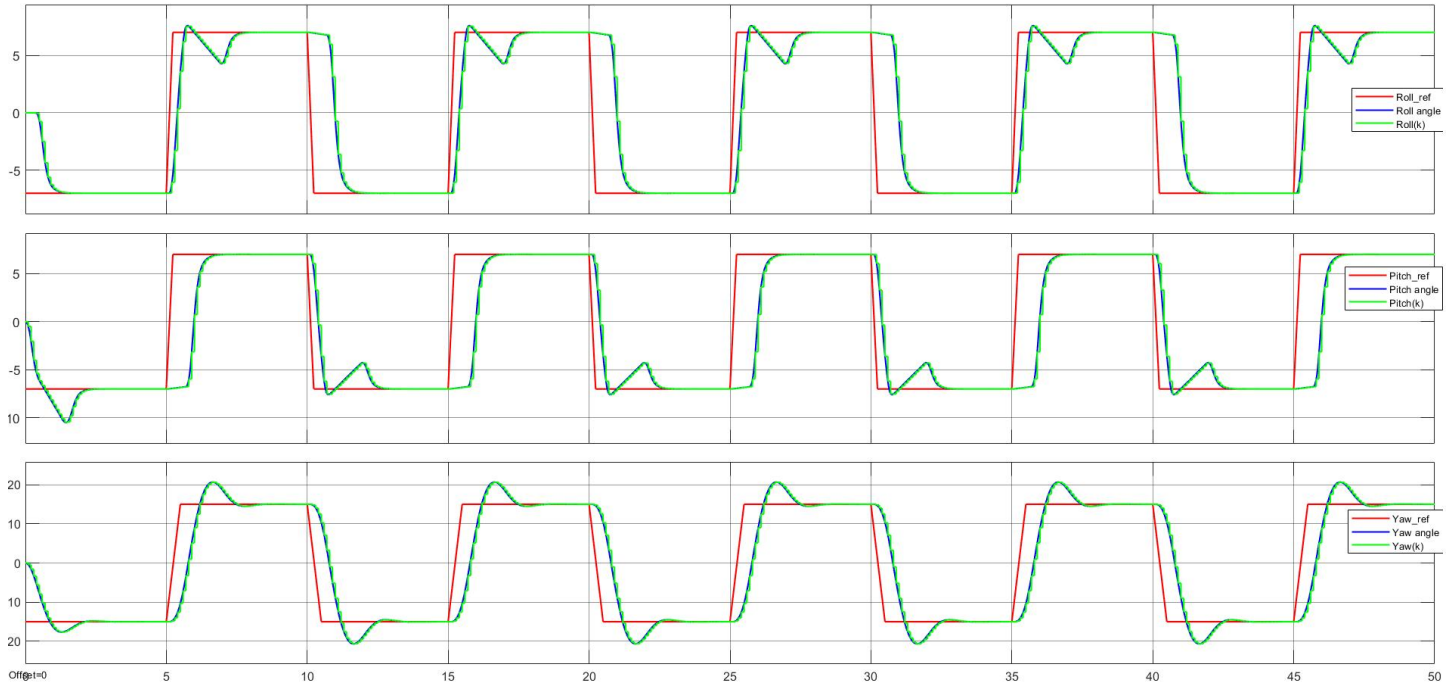


Figure 3.13: Simulated response of TU discretized system with T=0.1.

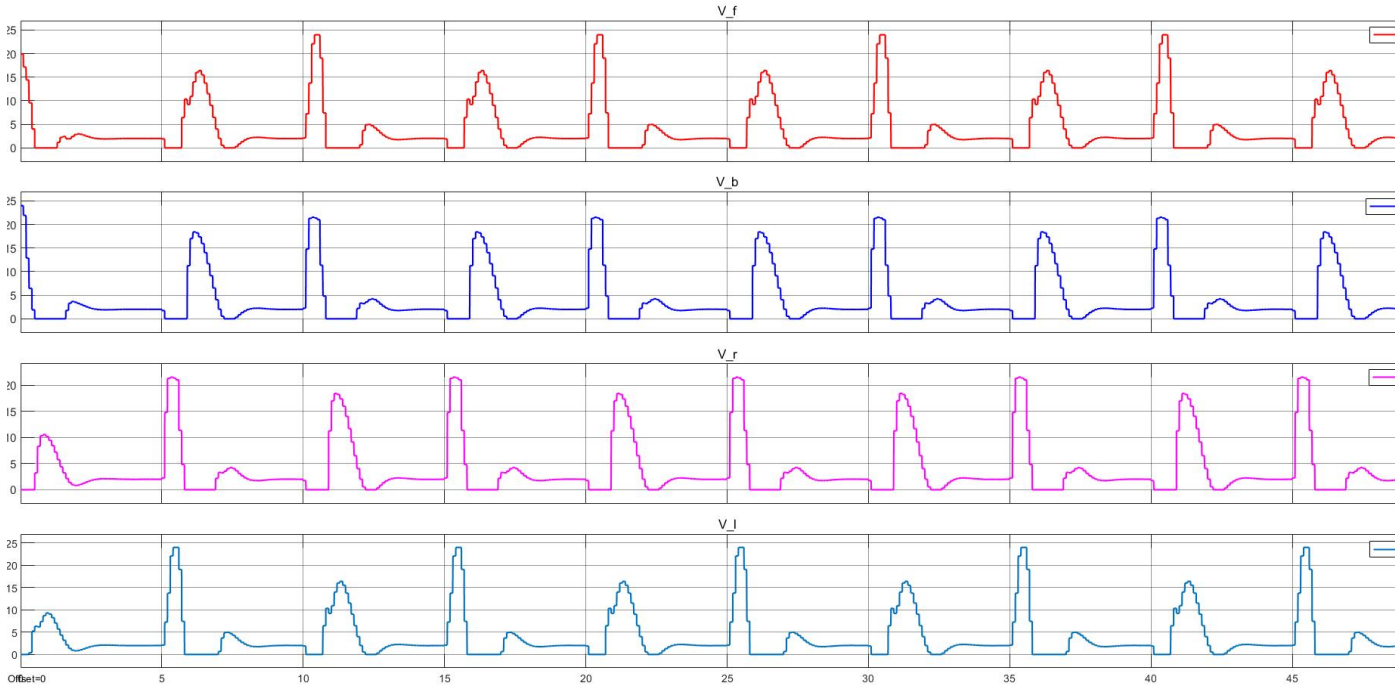


Figure 3.14: Simulated response of TU discretized system with T=0.1.

Chapter 4

Control

In the previous chapter it became clear that choice of discretization method plays a big part in the properties of a DT model. In this chapter the different models will be applied to design feedback control laws for the hover system and additional comparisons will be made.

4.1 Controllability

Controllability is an essential property of a dynamic system which describes its ability to move from any initial condition $x(0) = 0$ to any final state $x(k_f) = x_f$ by manipulation of its input vector u. The algebraic controllability theorem presented next provides a necessary and sufficient criterion to determine the controllability of a LTI system.

Theorem 4.1. *The discrete-time LTI system:*

$$x(k+1) = \mathbf{A}x(k) + \mathbf{B}u(k)$$

is controllable if and only if the controllability matrix $\mathcal{C}$:

$$\mathcal{C} = [\mathbf{B} \;\; \mathbf{AB} \;\; \mathbf{A}^2\mathbf{B} \;\; \cdots \;\; \mathbf{A}^{n-1}\mathbf{B}]$$

has full rank.

The $n \times nm$ matrix $\mathcal{C}$ has full rank (i.e. $rank(\mathcal{C}) = n$) if it contains n linearly independent columns or rows. Considering a fundamental result in linear algebra proves the column and row rank of any matrix are always equal, the controllability can be determined calculating either one[13].

In the case of the 3DOF hover system the controllability matrix $\mathcal{C}$ is a large 6×24 matrix, however, as shown in 2.3.5 matrix $\mathbf{A}$ is a nilpotent matrix with index $k = 2$ rendering $\mathbf{A}$ to the power of 2 and above equal to zero:

$$\mathcal{C} = [\mathbf{B} \;\; \mathbf{AB} \;\; \mathbf{A}^2\mathbf{B} \;\; \mathbf{A}^3\mathbf{B} \;\; \mathbf{A}^4\mathbf{B} \;\; \mathbf{A}^5\mathbf{B}] = [\mathbf{B} \;\; \mathbf{AB} \;\; \cdots \;\; 0]$$

$$\Downarrow$$

$$\mathcal{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{LK_f}{J_p} & -\frac{LK_f}{J_p} & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{LK_f}{J_r} & -\frac{LK_f}{J_r} & \cdots & 0 \\ 0 & 0 & 0 & 0 & -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} & & \vdots \\ \frac{LK_f}{J_p} & -\frac{LK_f}{J_p} & 0 & 0 & 0 & 0 & 0 & 0 & & \vdots \\ 0 & 0 & -\frac{LK_f}{J_r} & \frac{LK_f}{J_r} & 0 & 0 & 0 & 0 & \cdots & 0 \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{4.1}$$

Calculating the rank of the controllability matrix in Matlab yields:

$$rank(\mathcal{C}) = 6$$

proving that the system is indeed controllable. Furthermore, the controllability is preserved under discretization as shown in [14].

4.2 Linear Quadratic Regulator

A popular approach to designing a controller for a linear time-invariant (LTI) system is the linear quadratic regulator (LQR). LQR is an optimal control technique based on the state-space representation of a system. The technique is used to construct a linear feedback control law (4.2) that minimizes a linear cost function (4.3). Said cost function contains weight factors defined by the system engineer based on performance and energy requirements. Thus, the resulting control law provides the best possible performance with respect to predefined performance metrics.

$$u(k) = \mathbf{K}x(k) \tag{4.2}$$

When designing a controller for the 3DOF hover the objective is to reach the desired system state, $x_{ref}(k)$ as quickly as possible. When the desired state is obtained the error, $e(k) = x_{ref}(k) - x(k)$ becomes zero, this is achieved by driving the current state

$x(k)$ towards the reference through actuation given by the LQR control law. A discrete quadratic cost function (4.3) is chosen to quantify how well the desired state is reached:

$$J = \sum_{k=0}^{\infty} \left[e^T(k)\mathbf{Q}e(k) + u^T(k)\mathbf{R}u(k) \right] \quad [15] \tag{4.3}$$

To properly gauge the performance it is important to ensure all summed terms contribute positive to the cost, J. Therefore the $\mathbf{Q}$ matrix that penalizes poor performance must be a positive semi-definite matrix, $\mathbf{Q} \succcurlyeq 0$ such that $e^T(k)\mathbf{Q}e(k) \geq 0$ for any $e(k)$. A cost function containing only the first term is sufficient to bring the state error to zero as quickly as possible, however the calculated gain matrix would result in unphysical levels of actuation. The second term of the cost function is therefore added to penalize actuator usage. The positive definite $\mathbf{R}$ matrix act alongside $\mathbf{Q}$ as to provide a trade-off between quick performance and expended energy, without this the most effective control law would be found to have an infinitely large control signal $u(k)$.

A common method of choosing $\mathbf{Q}$ and $\mathbf{R}$ is to set them as diagonal $n \times n$ and $p \times p$ matrices, corresponding to the dimension of the state- and input vectors respectively. The i-th element along the diagonal represents the weight of the i-th state or input. For the hover, the following matrices were chosen:

$$\mathbf{Q} = \begin{bmatrix} 500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 350 & 0 & 0 & 0 & 0 \\ 0 & 0 & 350 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 \end{bmatrix} \tag{4.4}$$

$$\mathbf{R} = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix} \tag{4.5}$$

Once chosen the LQR problem can be solved to get the control law $u = -\mathbf{K}x(k)$ that produces the lowest cost, where $\mathbf{K}$ is given by:

$$\mathbf{K} = \left(\mathbf{R} + \mathbf{B}^T\mathbf{P}\mathbf{B} \right)^{-1} \mathbf{B}^T\mathbf{P}\mathbf{A} \tag{4.6}$$

and $\mathbf{P}$ is the stabilizing solution to the discrete-time algebraic Riccati equation (DARE)[15]:

$$\mathbf{P} = \mathbf{A}^T\mathbf{P}\mathbf{A} - \mathbf{A}^T\mathbf{P}\mathbf{B}\left(\mathbf{R} + \mathbf{B}^T\mathbf{P}\mathbf{B}\right)^{-1}\mathbf{B}^T\mathbf{P}\mathbf{A} \tag{4.7}$$

as mentioned a discrete quadratic cost function (4.3) is used to quantify the performance of the digital control. This is opposed to a continuous cost function (4.8). For small sampling times the difference is negligible, but as T grows interesting phenomena such as hidden oscillations/intersample ripple may occur which can skewer the resulting cost value, J.

$$J = \int_0^\infty \left[e^T(t)\mathbf{Q}e(t) + u^T(t)\mathbf{R}u(t)\right] \tag{4.8}$$

Figure 4.1 and 4.2 shows the real cost and the cost calculated using the discrete-time cost function (4.3).

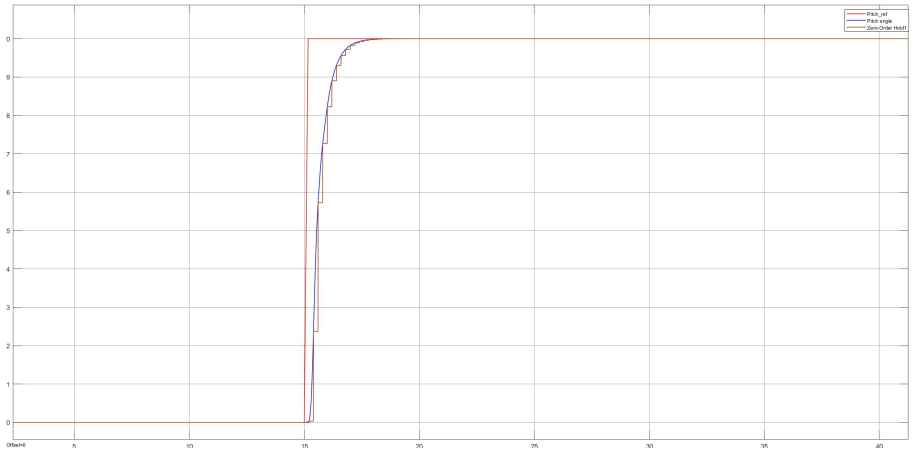


Figure 4.1: With a smaller sampling time of 0.2s the discrete-time and the continuous-time cost function yield very similar results for performance as expected

Increasing the sample time, T increases the cost as expected, but the different cost functions give misleading results as the intersample behaviour is neglected in the discrete-time version as made clear by Figure 4.2.

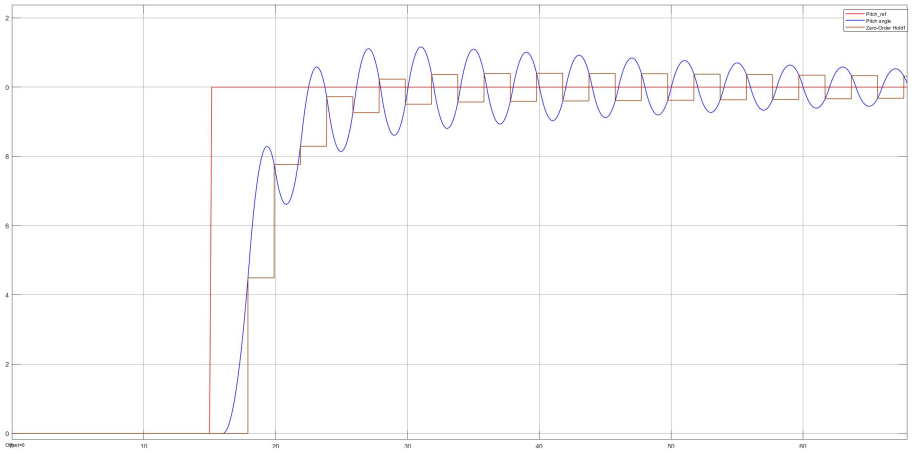


Figure 4.2: For a larger sampling time of 2s the discrete-time cost function results in a significantly smaller cost. The reason is missing information concerning intersample behaviour.

4.2.1 LQR Stability

For a system linearized about an equilibrium point x^* is asymptotically stable about x^* if solutions that start close enough to the point, x^* remain close to it and eventually converge to the equilibrium.

Given the infinite horizon LQR problem where $\mathbf{R} \succ 0$ is positive definite, $\mathbf{Q} \succcurlyeq 0$ is positive semi-definite and the pair $(\mathbf{A}, \mathbf{B})$ is observable and controllable there exists a unique and positive definite solution to the discrete algebraic riccati equation, P, and the optimal closed-loop system:

$$\dot{x} = (\mathbf{A} - \mathbf{BK})x$$

where:

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$$

is asymptotically stable. This can be verified by looking at the closed-loop eigenvalues of the system. If the real parts of the eigenvalues are negative it is a sufficient and necessary condition of a stable system as shown in section 3.1.

4.3 Simulated and applied results

Figure 4.3 display the systems simulated response to zero-order hold discretization and setpoint regulation for each axis. Figure 4.4 shows the actual applied response when the same settings were applied to the real 3DOF hover.

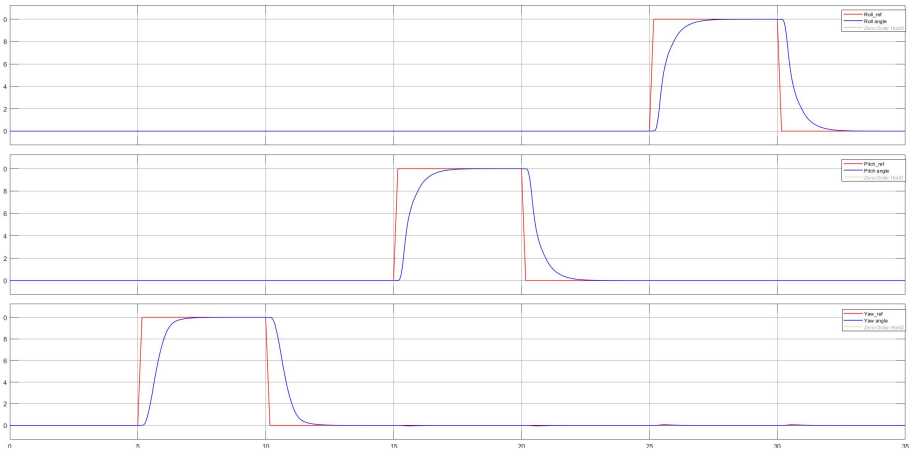


Figure 4.3: Simulated response of ZOH discretized system with T=0.2.

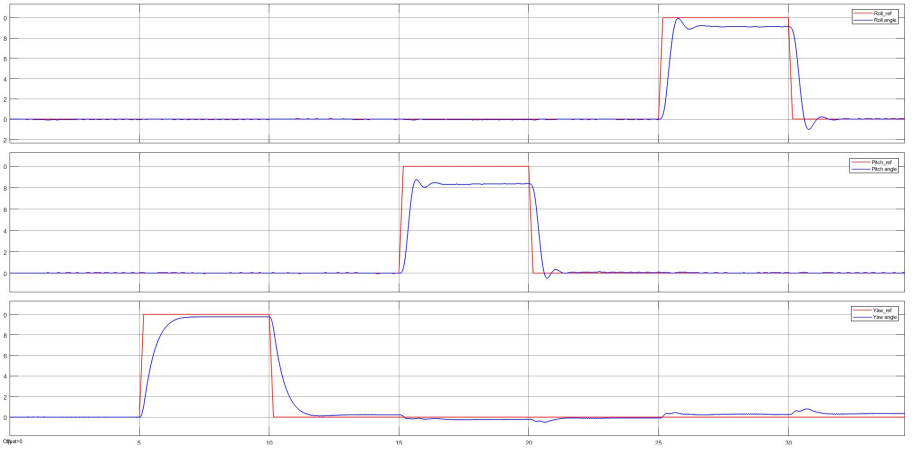


Figure 4.4: Actual response of ZOH discretized system with T=0.2 when applied to the hover system. The closed-loop response follows the reference quite well, however there is a static deviation due to the lack of integrator in the LQR method.

4.4 Event-triggered control

!! Why it is important to constrain energy consumption, communication and computation events !!

It is important to consider energy consumption and communication loads on systems. Rather than computing the control law (4.2) every constant time-step T like in section 4.2 u can be computed only when a certain threshold is reached where performance is no longer satisfactory. To define such a threshold a way to quantify performance is needed, one way to do this is by employing a Lyapunov function for the ideal closed-loop system.

The Lyapunov function can be thought of as a way of describing the energy of a system. One can imagine $V(x)$ as a set of contour lines with constant distance to the equilibrium point of a system x^* where its energy is equal to zero. The time derivative of the function $\dot{V}(x)$ describes the evolution of the total energy in the system *i.e.* whether it accumulates or disperses energy. Therefore, for a stable system the Lyapunov function must satisfy the following criteria:

$$\begin{cases} V(x) > 0, & \forall \ x \neq x^* \\ V(x^*) = 0 \end{cases}$$

$$\begin{cases} \dot{V}(x) < 0, & \forall \ x \neq x^* \\ \dot{V}(x^*) = 0 \end{cases}$$

Finding the Lyapunov function can be difficult in certain cases, but for a linear system $\dot{x} = (\mathbf{A} + \mathbf{BK})x$ the function is denoted by:

$$V(x(t)) = x^T(t)\mathbf{P}x(t) \tag{4.9}$$

Where $\mathbf{P} \in \mathbb{R}^{n \times n}$ is some symmetric and positive definite matrix. The derivative of $V(x)$ needs to satisfy the linear matrix inequality (LMI):

$$\begin{aligned} \dot{V}(x(t)) &= -x^T(t)\mathbf{Q}x(t) \\ &\leq -\sigma x^T(t)\mathbf{Q}x(t) \end{aligned} \tag{4.10}$$

Where $\sigma \in [0, 1]$ determines the rate of convergence of $\dot{V}(x)$. If a slow decrease of $\dot{V}(x)$ can be tolerated, σ can be set to a small value.

The time derivative of the Lyapunov function can be found by taking the partial derivatives of $V(x(t))$ along with the property $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T \Rightarrow \dot{x}^T(t) = x^T(t)(\mathbf{A} + \mathbf{BK})^T$:

$$\begin{aligned} \dot{V}(x(t)) = \frac{\partial V}{\partial x}\frac{\partial x}{\partial t} &= \dot{x}^T(t)\mathbf{P}x(t) + \underbrace{x^T(t)\dot{\mathbf{P}}x(t)}_{0} + x^T(t)\mathbf{P}\dot{x}(t) \\ &= x^T(t)\underbrace{\Big((\mathbf{A} + \mathbf{BK})^T\mathbf{P} + \mathbf{P}(\mathbf{A} + \mathbf{BK})\Big)}_{-\mathbf{Q}}x(t) \\ &= -x^T(t)\mathbf{Q}x(t) \leq -\sigma x^T(t)\mathbf{Q}x(t) \end{aligned} \tag{4.11}$$

Where the matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is symmetric and positive definite. The negativeness of the derivative $\dot{V}(x)$ guarantees the decreasing of the Lyapunov function.

Since the aim of event-triggered control is to reduce the amount of times the control input signal u is updated, the signal is held constant between consecutive updates using a zero-order hold.

$$u(t) = u(t_k), \quad \forall t \in [t_k, t_{k+1}), \quad k \in \mathbb{N}$$

Where the sequence $\{t_k\}$, $k \in \mathbb{N}$ is referred to as the triggering times and represent the instances at which the control signal (4.2) is updated and sent to the actuators. It is important to note that the triggering times will in general not be equidistant, therefore the implementation of event-triggered sampling will result in aperiodic control. The event-triggering condition is defined using equation (4.11), when the condition is about to be violated, *i.e.*, at the threshold where the inequality (4.11) becomes an equality the input signal is recomputed. Introducing the error variable:

$$e(t) = x(t_k) - x(t) \quad \forall t \in [t_k, t_{k+1}), \quad k \in \mathbb{N} \tag{4.12}$$

using (4.12), the closed-loop system during the time interval $[t_k, t_{k+1})$ can be rewritten as:

$$\begin{aligned} \dot{x} &= \mathbf{A}x(t) + \mathbf{B}Kx(t_k) \\ &= \mathbf{A}x(t) + \underbrace{\mathbf{BK}x(t) - \mathbf{BK}x(t)}_{0} + \mathbf{B}Kx(t_k) \\ &= \mathbf{A}x(t) + \mathbf{BK}x(t) + \mathbf{BK}\underbrace{(x(t_k) - x(t))}_{e(t)} \\ &= (\mathbf{A} + \mathbf{BK})x(t) + \mathbf{BK}e(t) \end{aligned} \tag{4.13}$$

By considering this into the time derivative of the Lyapunov function (4.11) it can be rewritten as:

$$\dot{V}(x) = -x^T(t)\mathbf{Q}x(t) + 2x^T(t)\mathbf{PBK}e(t) \tag{4.14}$$

Finally, the event-triggering condition can be defined by rewriting the inequality (4.11) in the following manner:

$$-x^T(t)\mathbf{Q}x(t) + 2x^T(t)\mathbf{PBK}e(t) \leq -\sigma x^T(t)\mathbf{Q}x(t)$$

$$\Downarrow$$

$$-(1-\sigma)x^T(t)\mathbf{Q}x(t) + 2x^T(t)\mathbf{PBK}e(t) \leq 0 \tag{4.15}$$

The event is triggered at the triggering times, t_k which is defined as the times when the equality (4.15) holds. The irregular occurrence of these instants proposes the question of the existence of a lower bound, $\tau, t_{k+1} - t_k \geq \tau, k \in \mathbb{N}$, which in the literature [16] is referred to as the minimal inter-event time. When the minimal inter-event time is too small, approaching or even equal to zero, an event triggered implementation will require very fast updates, even infinitely fast for $\tau = 0$. Such a value for τ would be impossible to implement digitally.

4.4.1 Simulated and experimental results

The simulated results worked out as expected after implementing the triggering condition (4.15) into simulink using function blocks as seen in Appendices B B and C C. The condition was designed to drive the system to zero as opposed to the error signal as for standard feedback controllers. After initiating the system the condition was violated quickly and performed as expected. The performance of the control and the actuator instances are presented in Figure 4.5.

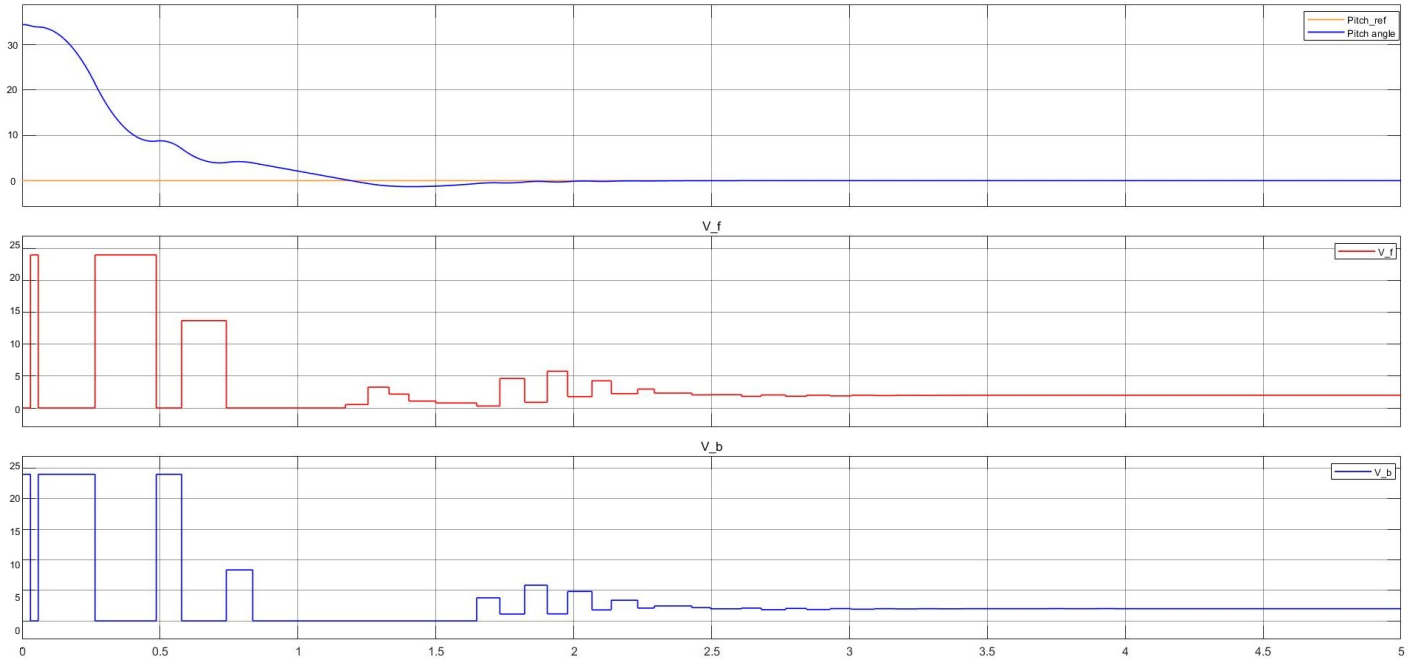


Figure 4.5: Simulated response of event-triggered control when initialized at an elevated pitch angle.

The same setup was implemented on the laboratory model, however the resulting performance was unsatisfactory. Despite the triggering condition performing as expected as seen in Figure 4.6.

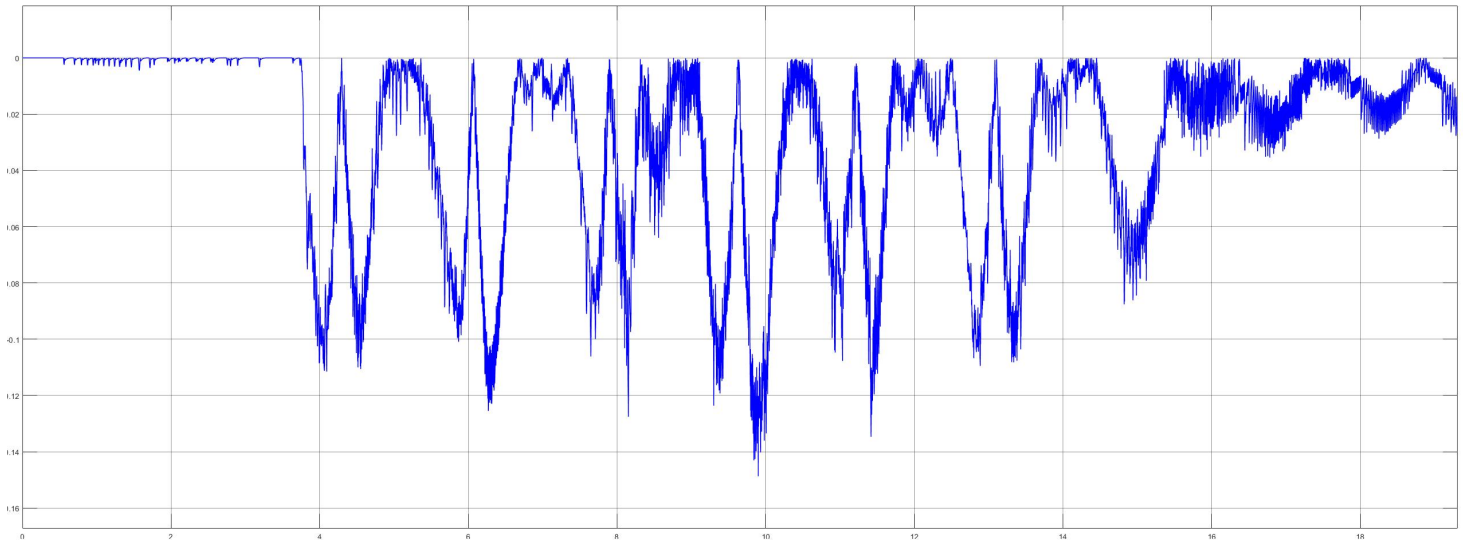


Figure 4.6: Trigger condition as deviations from zero was introduced through pushing the model.

Figure 4.7 shows the motors being actuated as the hover deviates from zero.

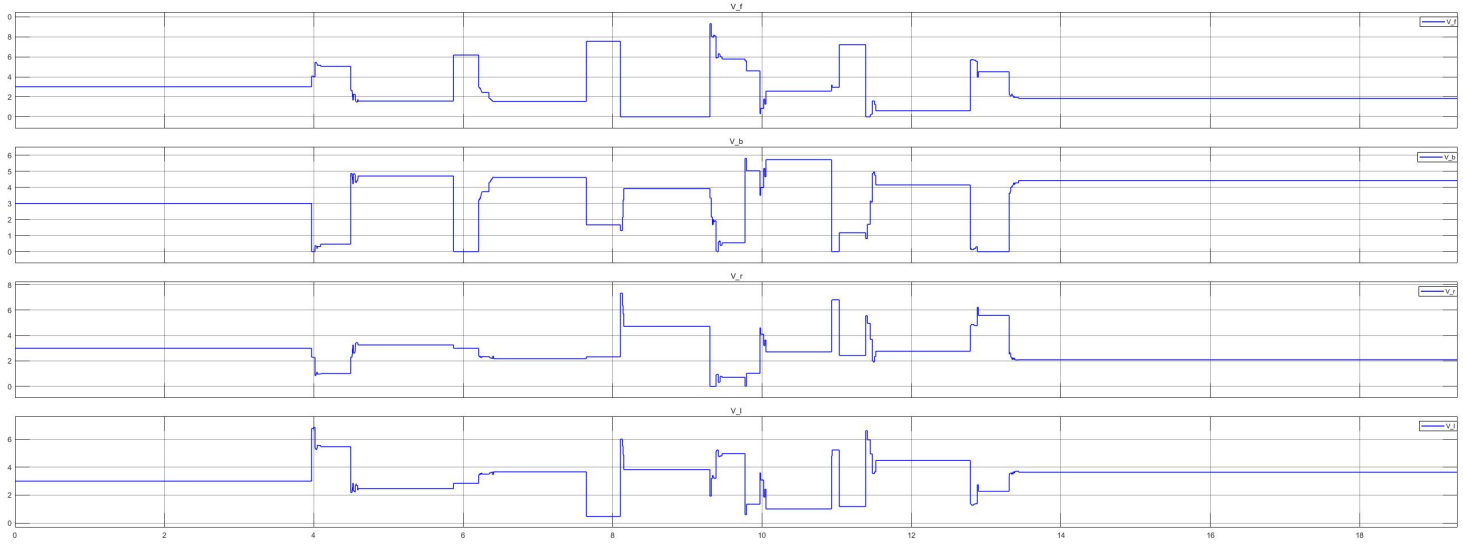


Figure 4.7: Motor actuation during experiments on the hover system as the trigger condition is violated. The asynchronous samples are clearly present.

Chapter 5

Conclusion and Future Directions

The foundation of this thesis was to investigate sampled-data strategies for control of a continuous- time system. The main goal was to compare different discretization techniques, sampling times and strategies to find more optimal and efficient ways of performing digital system control. In conclusion the forward Euler discretization technique is the worst method since it does not preserve stability and has lower performance in most cases then the other techniques reviewed. The backwards Euler performes better however it does not compare very well to the Tustin approximation which in turn can not compete against the zero-order hold. This was expected as the ZOH model is a mathematical model of the practical signal reconstruction done by conventional digital to analog converter. It was also possible to control the system with much larger sampling times which would be beneficial in resource constrained environments. The event-triggered control unfortunately didnt go as well as i had hoped, but the simulation showed promise. The triggering function picked up when the system deviated past a certain threshold and the controller was updated (however poorly so).

Appendix A

Thesis poster

Sampled Data Strategies For The Control of a 3DOF Hover System

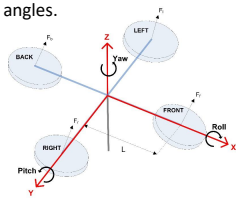
Emil E. Kleivdal
University of Stavanger

The System

A sampled data control system refers to a control system in which a continuous plant interacts with a digital controller.

The hover consists of a round frame with four propellers, each driven by a DC motor. The hover is mounted on a 3 degrees of freedom (DOF) pivot joint. The propellers generate lift that can be used to move control the pitch, roll and yaw angles.

The position of the hover is monitored by three encoders, one on each axis. These encoders produce a continuous signal that forms the basis of the state-space model used to mathematically represent the hover system.



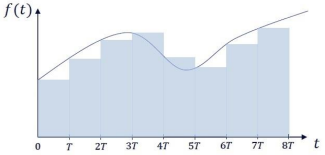
Discretization

For a digital controller to interact with the continuous-time signals coming from the hover, said signals must first be transformed into their discrete-time counterparts. Several discretization techniques are utilized in this thesis to achieve this.

- The explicit Euler's method , as seen in the figure below.
- The implicit Euler's method.
- Tustin approximation, sometimes referred to as the bilinear transform.
- Zero order hold discretization.

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \Longrightarrow \begin{cases} x\left((k+1)T\right) = A_d^{FE}x(kT) + B_d^{FE}u(kT) \\ y(kT) = C_d^{FE}x(kT) + D_d^{FE}u(kT) \end{cases}$$

$$\mathbf{A}_d^{FE} = \mathbf{I} + \mathbf{A}$$
$$\mathbf{B}_d^{FE} = T\mathbf{B}$$
$$\mathbf{C}_d^{FE} = \mathbf{C}$$
$$\mathbf{D}_d^{FE} = \mathbf{D}$$



Modelling

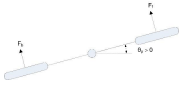
The dynamics of each axis can be described by the following three equations:

$$\ddot{\theta}_p = \frac{LK_f}{J_p}(V_f - V_b)$$
$$\ddot{\theta}_r = \frac{LK_f}{J_r}(V_r - V_l)$$
$$\ddot{\theta}_y = \frac{K_t}{J_y}(V_r + V_l) + \frac{K_t}{J_y}(V_f + V_b)$$

The roll and pitch models are similar as the angle is decided by the difference in thrust between the two propellers on said axis, as the figure below shows. The yaw angle on the other hand is determined by all four propellers together.

Symbol	Description	Value	Unit
J_p	Moment of inertia about pitch axis	0.0552	$Kg - m^2$
L	Distance from propeller to pivot point	0.197	m
K_f	Force-thrust constant of motor/propeller	0.1188	N/V
θ_p	Pitch angle	–	Rad
V_f	Voltage applied to front motor	–	V
V_b	Voltage applied to back motor	–	V

Table 2.1: Variables regarding pitch model.



From these equations a state-space model given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

can be constructed by combining a proper choice of state, output and input vectors with the equations for angular motion above.

$$\mathbf{x}^T = \begin{bmatrix} \theta_p & \theta_r & \theta_y & \dot{\theta}_p & \dot{\theta}_y & \dot{\theta}_y \end{bmatrix}$$
$$\mathbf{y}^T = \begin{bmatrix} \theta_p & \theta_r & \theta_y \end{bmatrix}$$
$$\mathbf{u}^T = \begin{bmatrix} V_f & V_b & V_l & V_r \end{bmatrix}$$

Symbol	Description
$\mathbf{x} \in \mathbb{R}^n$	State vector
$\mathbf{y} \in \mathbb{R}^q$	Output vector
$\mathbf{u} \in \mathbb{R}^p$	Input vector
$\mathbf{A} \in \mathbb{R}^{n \times n}$	System matrix
$\mathbf{B} \in \mathbb{R}^{n \times p}$	Input matrix
$\mathbf{C} \in \mathbb{R}^{q \times n}$	Output matrix
$\mathbf{D} \in \mathbb{R}^{q \times n}$	Feedthrough matrix

The resulting state-space model written out was found to be:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \\ \dot{x}_5(t) \\ \dot{x}_6(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_6(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{LK_f}{J_p} & -\frac{LK_f}{J_p} & 0 & 0 \\ 0 & 0 & -\frac{LK_f}{J_r} & \frac{LK_f}{J_r} \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} \end{bmatrix}}_{\mathbf{B}} \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ u_4(t) \end{bmatrix}$$

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_6(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{D}} \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ u_4(t) \end{bmatrix}$$

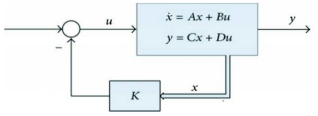
Linear Quadratic Regulator

A popular approach to designing a controller for a linear time-invariant (LTI) system is the linear quadratic regulator (LQR). LQR is an optimal control technique based on the state-space representation of a system.

The technique is used to design a control law, $u = \mathbf{K}x$ that minimizes some cost function, J.

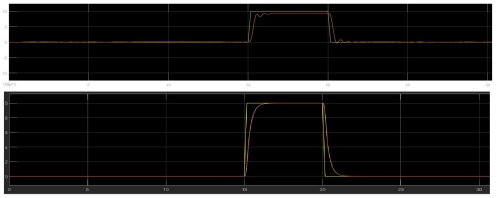
$$J = \sum_{k=0}^{\infty} \left[e^T(k)\mathbf{Q}e(k) + u^T(k)\mathbf{R}u(k) \right]$$

The cost function includes matrices **Q** and **R** that penalizes the time it takes to regulate as well as the energy expenditure during regulation. Tweaking the matrices allows for additional freedom when tailoring the regulator to a specific system.

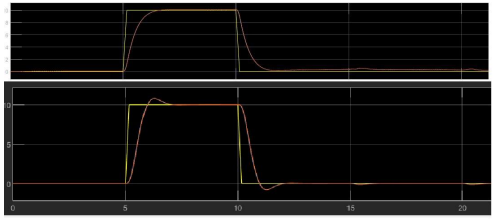


Example of a LQR controlled system.

The following graph demonstrates the performance of LQR control applied to the hover system compared to simulations after zero order hold discretization:



Graph 1, Pitch angle comparison.



Graph 2, Yaw angle comparison.

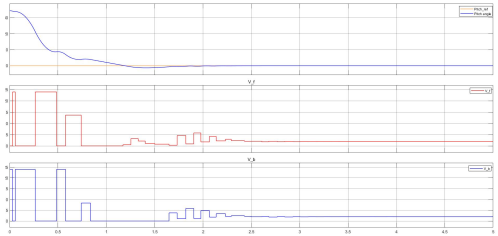
Event triggered control

The idea is to monitor the state of a system and only compute and transmit output signals when it is necessary. The aim is to reduce communication load on networked systems as well as the computational power required by embedded microprocessors.

This is accomplished by implementing an event triggering condition.

$$-(1-\sigma)x^T(t)\mathbf{Q}x(t) + 2x^T(t)\mathbf{PBK}e(t) \leq 0$$

The condition is violated once the system deviates from the desired state past a certain threshold. The figure below shows how the event triggered system brings back to zero after, for instance a gust of wind. Note that the control law is not updated with a fixed period.



A drawback of event triggered control is that it requires the constant monitor of the systems states, a possible way past this is self triggered control. In self triggered control measurements are only taken at certain sampling instants, where at each sampling instant the next is determined.

This makes it possible to completely shut off sensors and communication between samples, further reducing energy usage and bandwidth requirements, however it renders the system strictly open-loop between samples.

Acknowledgements

A special thank you to Damiano Rotondo for invaluable help with the project.

Appendix B

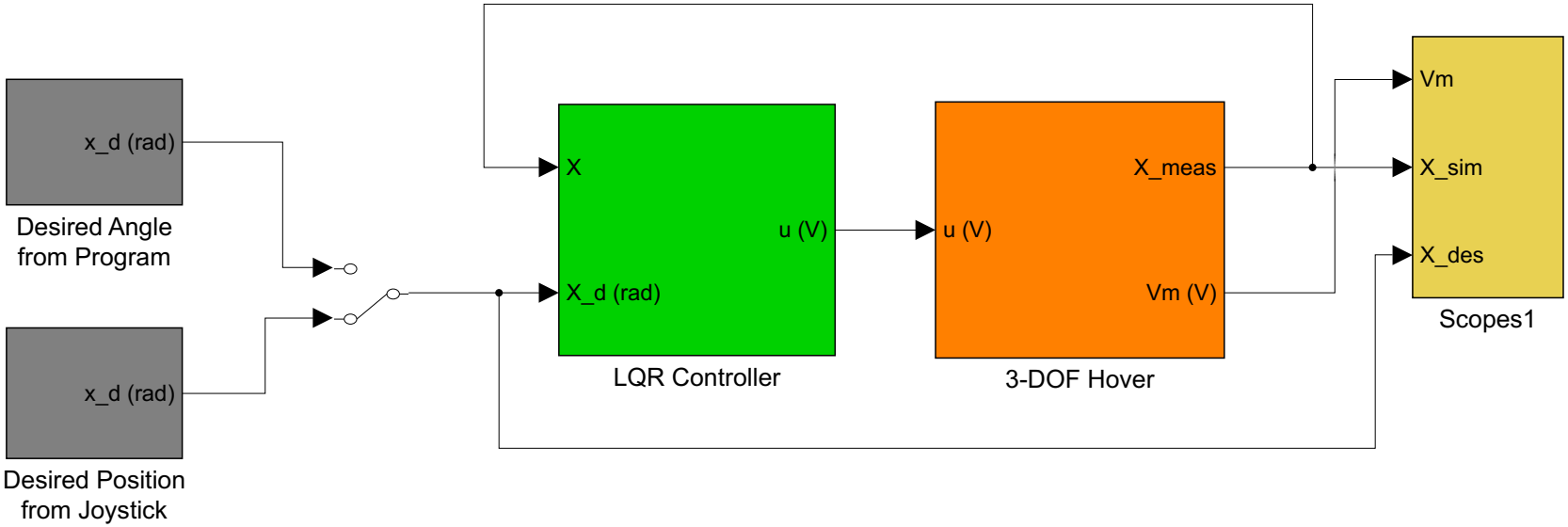
MATLAB code

```
sigma = 0;
P = sdpvar(6);
Constraints = [P >= 1e-7*eye(6), (A-B*K)'*P + P*(A-B*K) <= -1e-07*eye(6)];
ops = sdpsettings('solver','mosek','verbose',0);
optimize(Constraints,[],ops);
P=double(P);
Q = -((A-B*K)'*P + P*(A-B*K));
filnavn = 'LMI_vars';
save(filnavn,'P','Q');
```

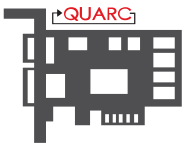
Appendix C

Simulink schemes

Quanser 3 DOF Hover:
Closed-loop Actual System



3DOF HOVER Subsystem
Reads angles from encoder and applied voltage to motors.



A0 #0 = front motor
A0 #1 = back motor
A0 #3 = right motor
A0 #2 = left motor

u = [u_front, u_back, u_right, u_left

1
u (V)

UPM Voltage Limit:
0 to VMAX_AMP

1 / K_AMP*u
Cable Gain
Pre-Compensation

HIL
Write
Analog
(HIL-1)

Motor Input Voltage

K_AMP*u
Cable Gain

2
Vm (V)

Ch #0 = pitch
Ch #1 = roll
Ch #2 = yaw

HIL
Read
Encoder
Timebase
(HIL-1)

0 pitch
1 roll
2 yaw

K_EC_Y
Encoder Resolution:
Yaw

K_EC_P
Encoder Resolution:
Pitch

K_EC_R
Encoder Resolution:
Roll

yaw
pitch
roll

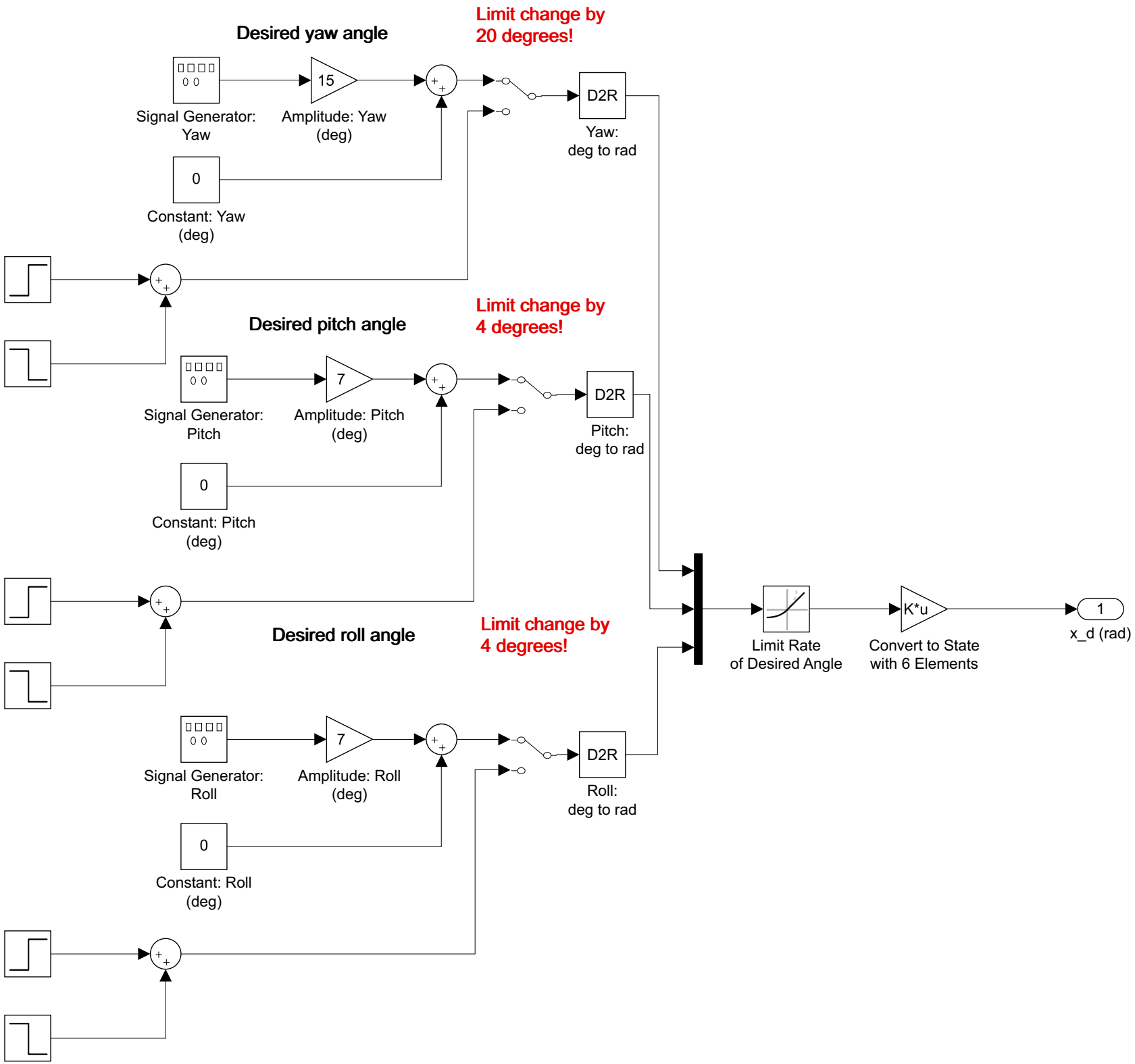
$\frac{num(s)}{den(s)}$
Derivative Filter: Yaw

$\frac{num(s)}{den(s)}$
Derivative Filter: Pitch

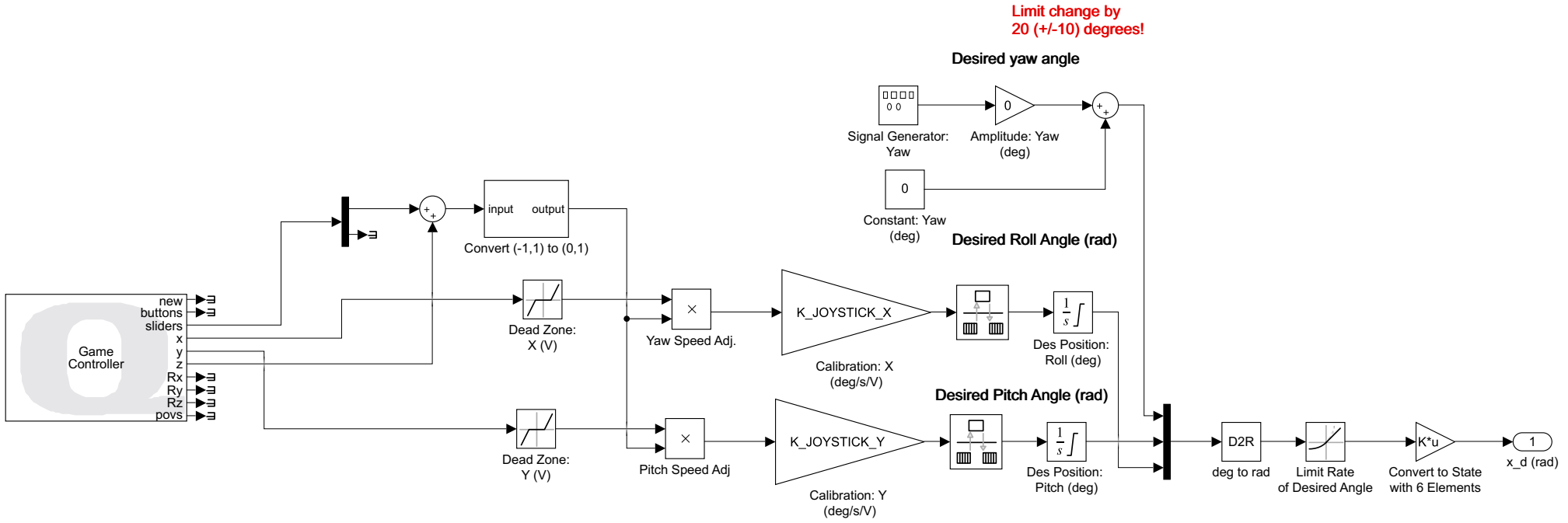
$\frac{num(s)}{den(s)}$
Derivative Filter: Roll

x = [theta_y, theta_p, theta_r, theta_dot_y, theta_dot_p, theta_dot_r,]

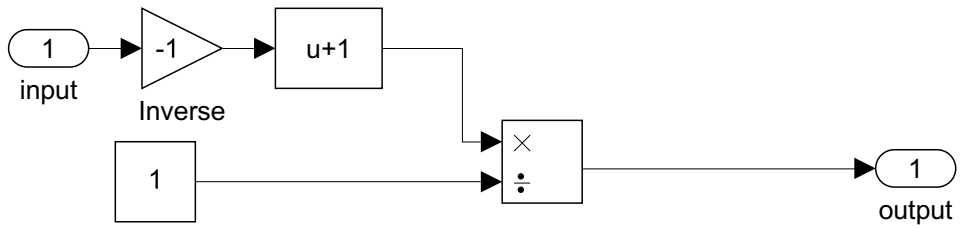
1
X_meas


Desired yaw angle
Limit change by
20 degrees!
Signal Generator:
Yaw
15
Amplitude: Yaw
(deg)
D2R
Yaw:
deg to rad
0
Constant: Yaw
(deg)
Desired pitch angle
Limit change by
4 degrees!
Signal Generator:
Pitch
7
Amplitude: Pitch
(deg)
D2R
Pitch:
deg to rad
0
Constant: Pitch
(deg)
Desired roll angle
Limit change by
4 degrees!
Signal Generator:
Roll
7
Amplitude: Roll
(deg)
D2R
Roll:
deg to rad
0
Constant: Roll
(deg)
Limit Rate
of Desired Angle
K*u
Convert to State
with 6 Elements
1
x_d (rad)

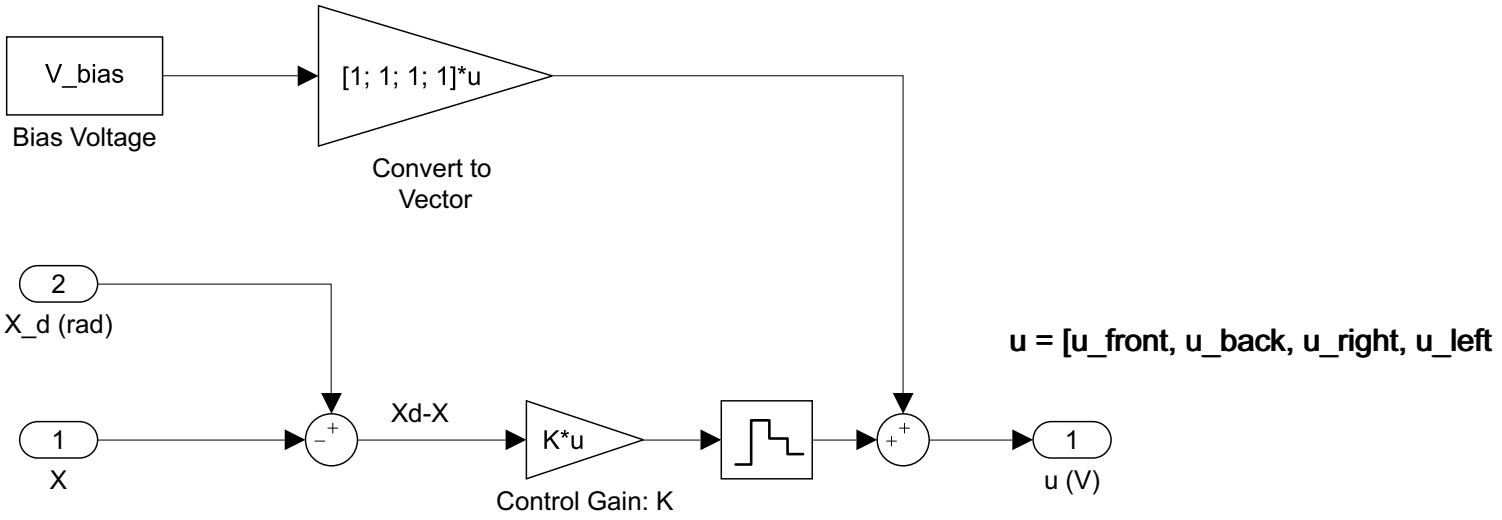
Joystick Setup for Commanding Angular Positions on Hover



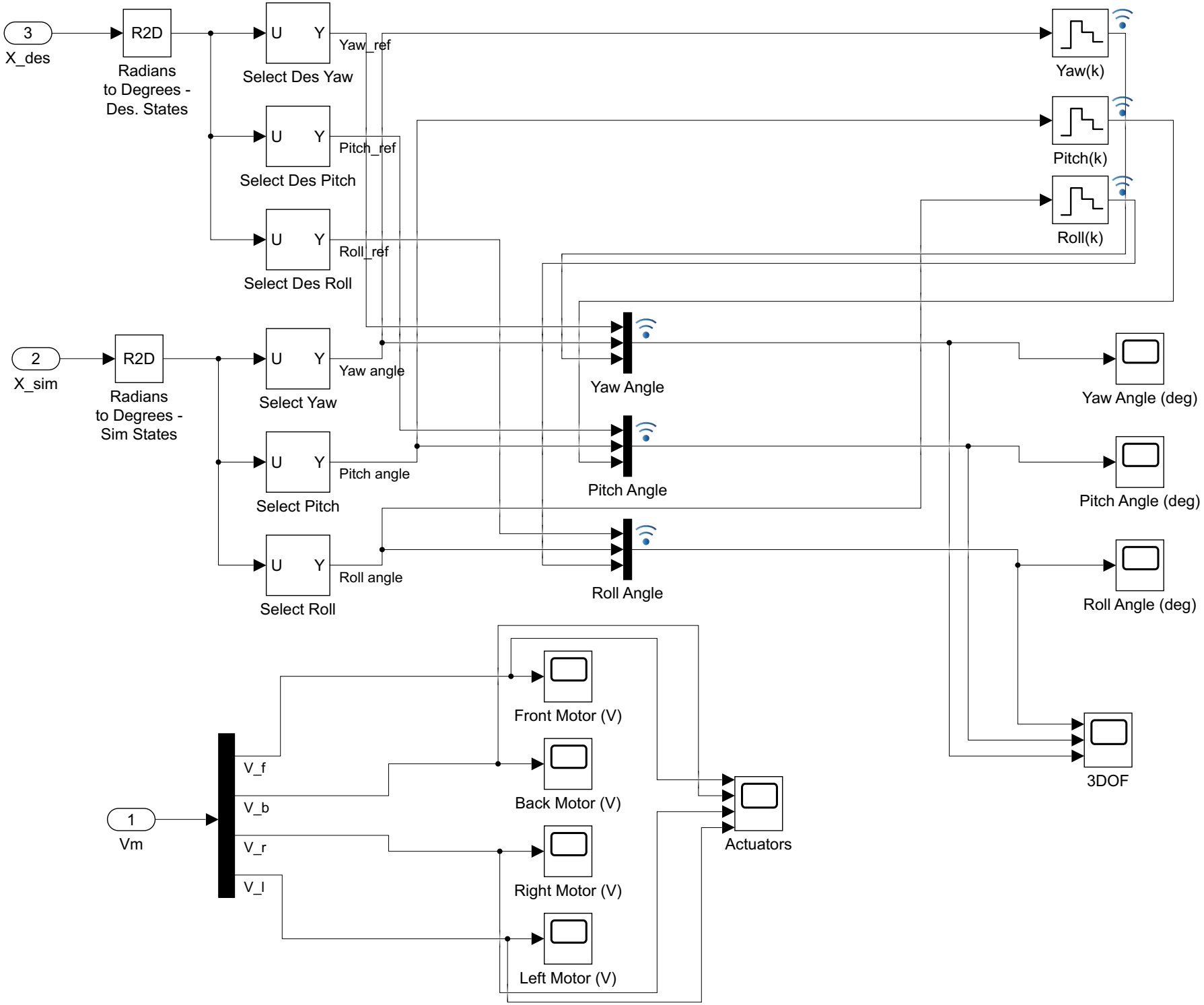
Inverse input and change range from (-1,1) to (0,1)



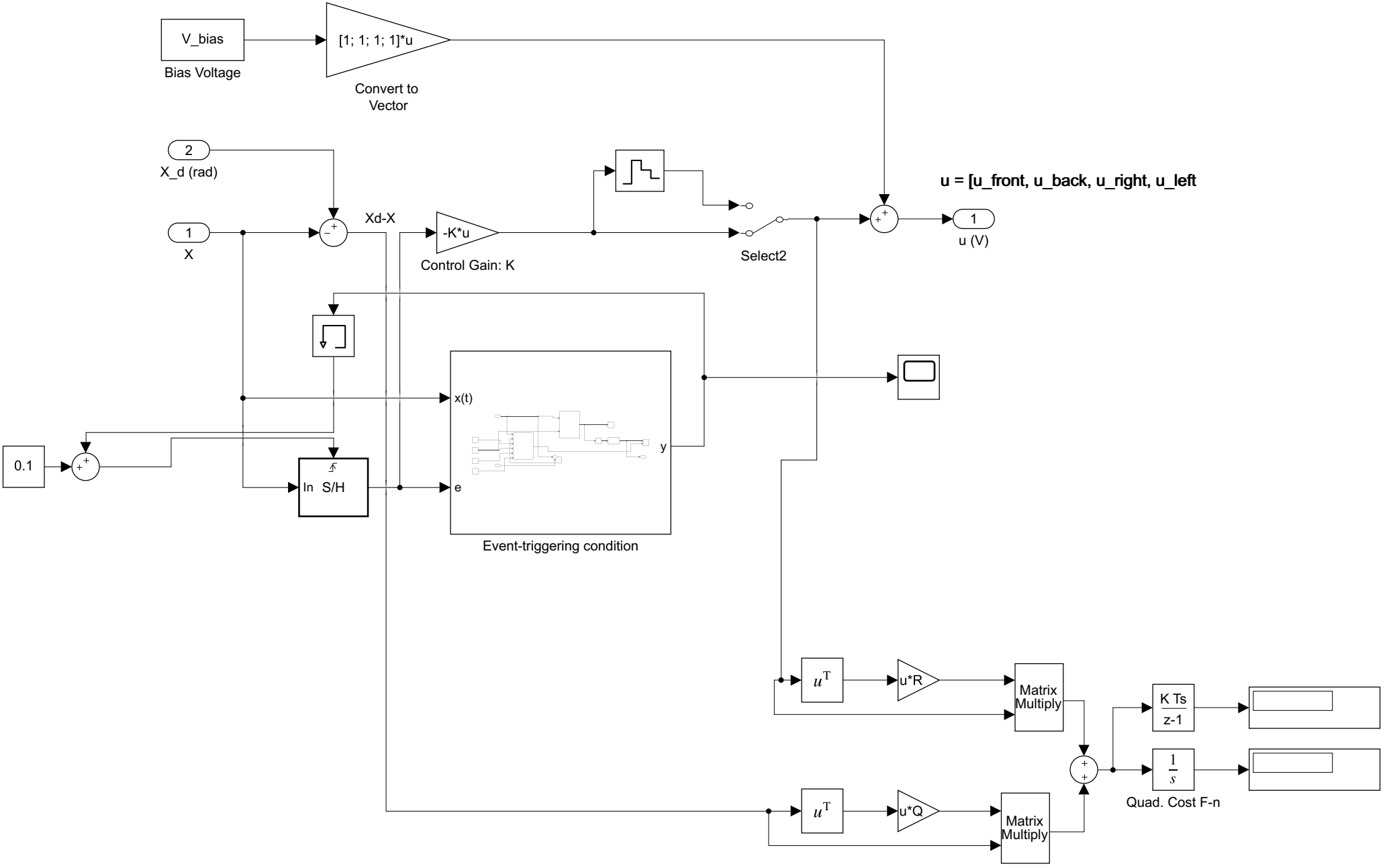
3DOF HOVER LQR+FF Controller

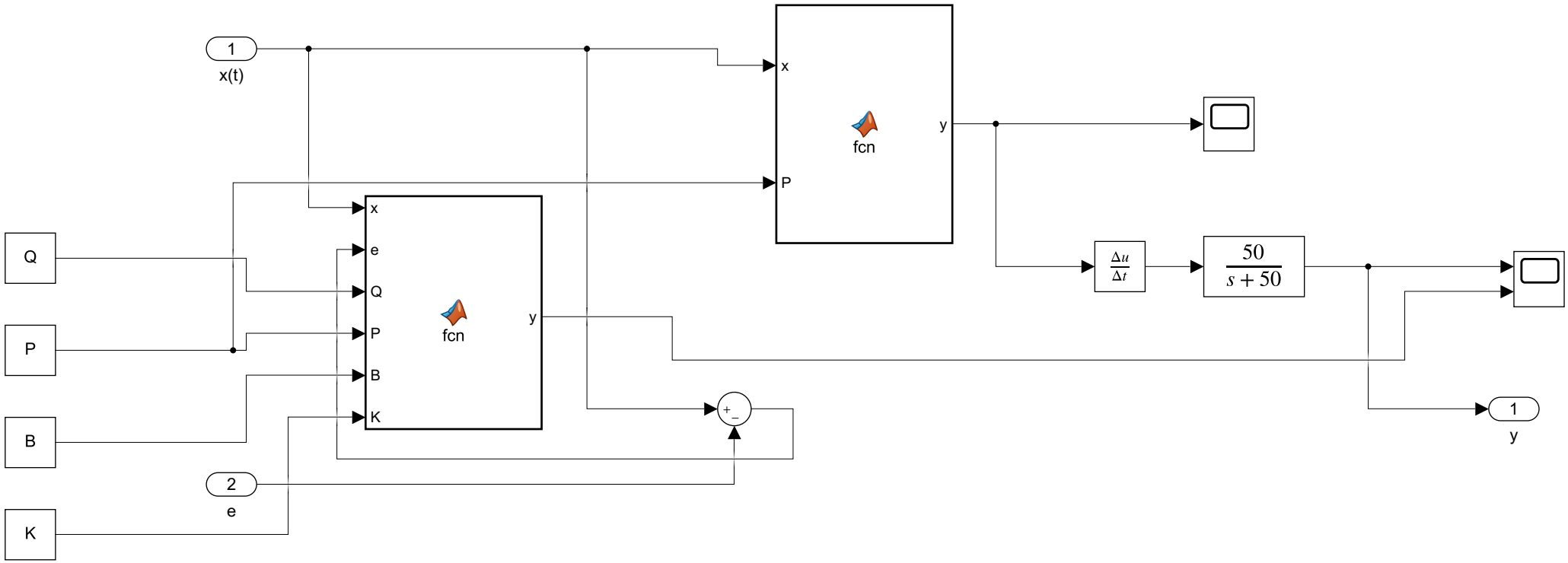


Scopes

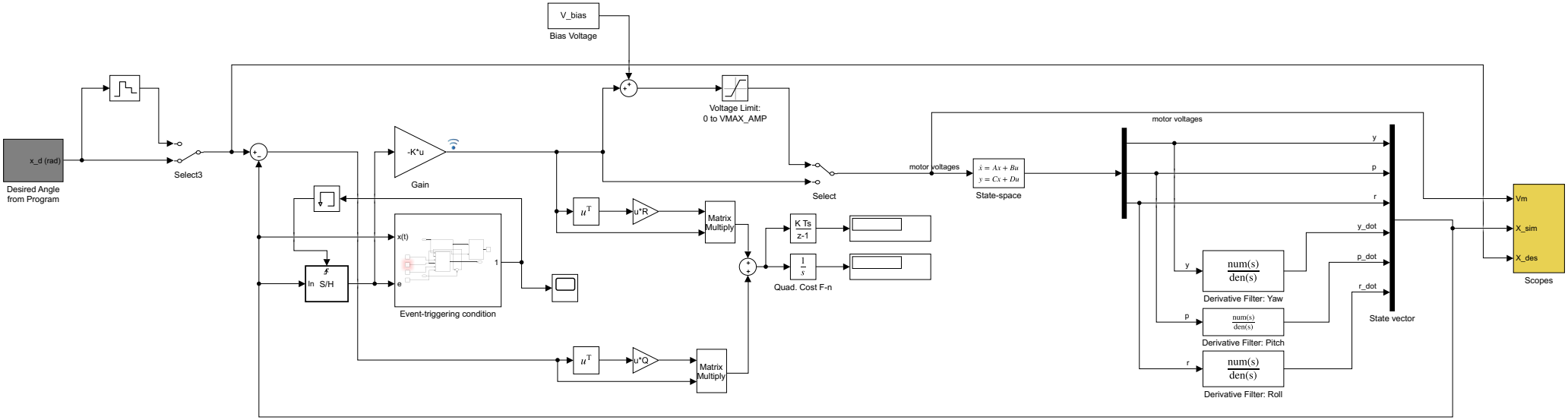

3
X_des
R2D
Radians
to Degrees -
Des. States
U
Y
Select Des Yaw
Yaw_ref
Select Des Pitch
Pitch_ref
Select Des Roll
Roll_ref
Yaw(k)
Pitch(k)
Roll(k)
2
X_sim
R2D
Radians
to Degrees -
Sim States
Select Yaw
Yaw angle
Select Pitch
Pitch angle
Select Roll
Roll angle
Yaw Angle
Pitch Angle
Roll Angle
Yaw Angle (deg)
Pitch Angle (deg)
Roll Angle (deg)
3DOF
1
Vm
V_f
V_b
V_r
V_l
Front Motor (V)
Back Motor (V)
Right Motor (V)
Left Motor (V)
Actuators

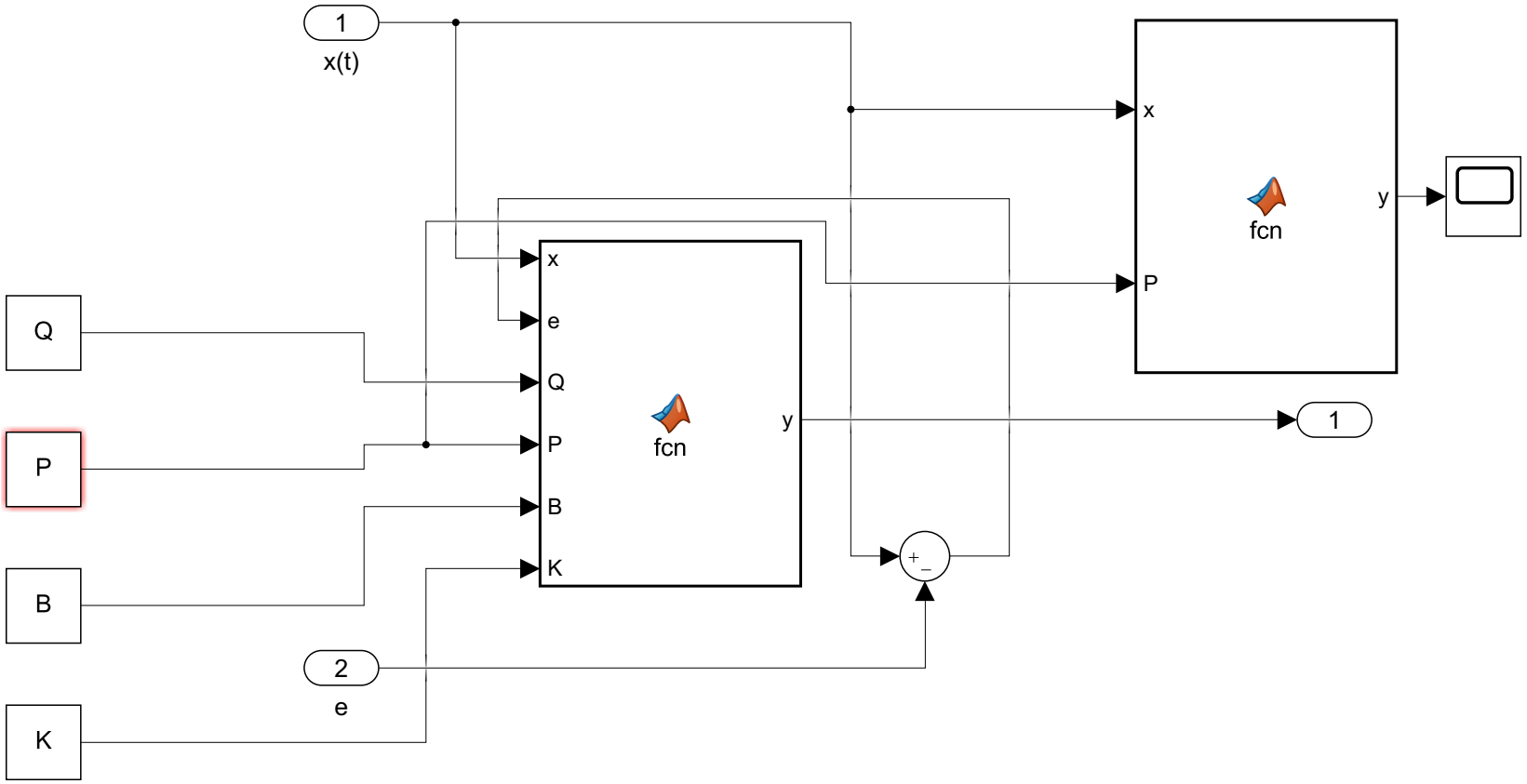
3DOF HOVER LQR+FF Controller



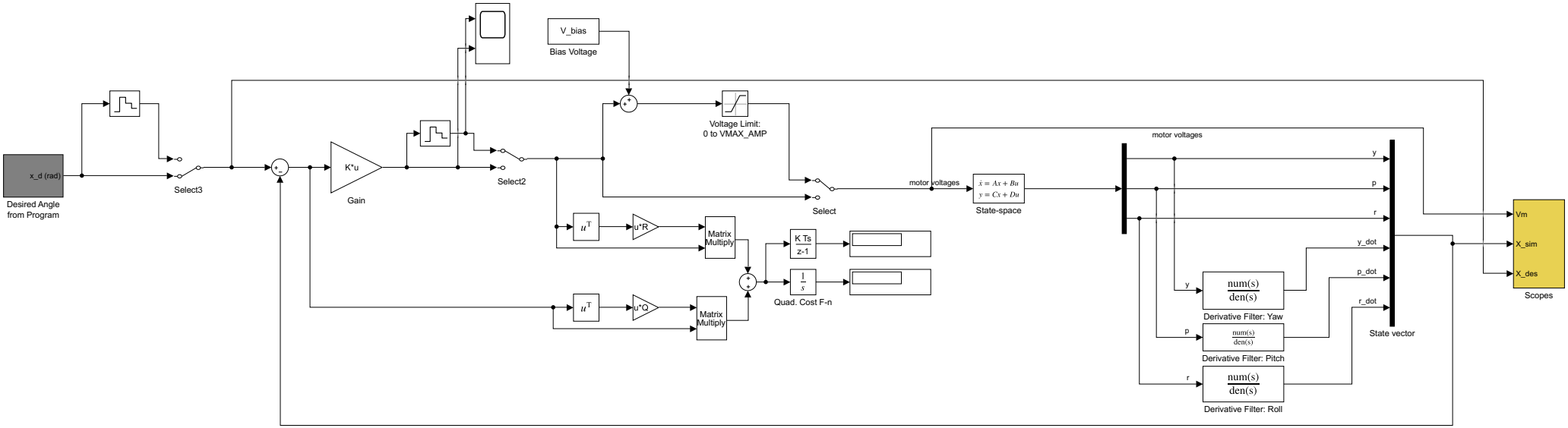

1
x(t)
x
P
fcn
y
x
e
Q
P
B
K
fcn
y
Q
P
B
K
2
e
+
−
Δu/Δt
50/(s + 50)
1
y

Quanser 3 DOF Hover: Simulation




1
x(t)
x
e
Q
P
B
K
fcn
y
1
2
e
Q
P
B
K
x
P
fcn
y
+
_

Quanser 3 DOF Hover: Simulation



List of Figures

List of Tables

Bibliography

[1] S. Gupta. Increasing the sampling efficiency for a control system. *IEEE Transactions on Automatic Control*, 8(3):263–264, 1963. doi: 10.1109/TAC.1963.1105568.

[2] A. Liff and J. Wolf. On the optimum sampling rate for discrete-time modeling of continuous-time systems. *IEEE Transactions on Automatic Control*, 11(2):288–290, 1966. doi: 10.1109/TAC.1966.1098317.

[3] G. Bekey and R. Tomovic. Sensitivity of discrete systems to variation of sampling interval. *IEEE Transactions on Automatic Control*, 11(2):284–287, 1966. doi: 10.1109/TAC.1966.1098309.

[4] R. Tomovic and G. Bekey. Adaptive sampling based on amplitude sensitivity. *IEEE Transactions on Automatic Control*, 11(2):282–284, 1966. doi: 10.1109/TAC.1966.1098308.

[5] D. Ciscato and L. Martiani. On increasing sampling efficiency by adaptive sampling. *IEEE Transactions on Automatic Control*, 12(3):318–318, 1967. doi: 10.1109/TAC.1967.1098605.

[6] J. Mitchell and W. McDaniel. Adaptive sampling technique. *IEEE Transactions on Automatic Control*, 14(2):200–201, 1969. doi: 10.1109/TAC.1969.1099144.

[7] M. Lévis J. Apkarian. Laboratory guide: 3 dof hover experiment. Quanser Inc.

[8] Hakim Bouadi and Félix Mora-Camino. Modeling and adaptive flight control for quadrotor trajectory tracking. *Journal of Aircraft*, 55:666–681, 2017.

[9] Damiano Rotondo. *Lecture notes Part III. Discretization.* 2021.

[10] Wikipedia. Trapezoidal rule. .

[11] Damiano Rotondo. *Lecture notes Part II. Elements of system theory.* 2021.

[12] Electronicscoach.com. *Stability of control system.*

[13] Wikipedia. Rank(linear algebra. .

[14] K.S. Narendra R.E Kalman, Y.C. Ho. *Controllability of liner dynamical systems.* pages 189–213, 1963.

[15] Damiano Rotondo. *Lecture notes Part IV. LQR control.* 2021.

[16] P. Tabuada. *Event-triggered real-time scheduling of stabilizing control tasks.* pages 1680–1685, 2007.