University of Stavanger

Faculty of Science and Technology

MASTER’S THESIS

Study program/ Specialisation: Robot Technology and Signal Processing, Master‘s Degree Programme	Spring semester 2023 Open/ ~~Restricted access~~

Writer: Vetle Normann

Faculty supervisor: Damiano Rotondo

External supervisor(s):

Thesis title:

Event-triggered and self-triggered control of a 3 DOF hover system

Credits (ECTS): 30

Keywords: Event-triggered control Self-triggered control Aperiodic control systems Quanser 3 DOF hover	Pages: 72 + enclosure: 29 Stavanger 15. June 2023

University of Stavanger

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

Event-triggered and self-triggered control of a 3 DOF hover system

Master's Thesis in Robot Technology and Signal Processing
by

Vetle Normann

Internal Supervisor

Damiano Rotondo

June 15, 2023

“Only dead fish go with the flow”

Andy Hunt,
Grandma’s quote

Abstract

Many modern control systems are implemented on devices that are resource constrained in the form of computational power, energy consumption or bandwidth. Control systems are conventionally implemented through periodic sampling and updating of the control action, referred to as periodic control systems. As a consequence, periodic control systems waste resources by updating the control action and sampling the state of a system, even when it is not necessary. A more efficient way of spending resources is only to update the control action or conduct sampling when necessary, referred to as aperiodic control systems.

This thesis presents the theoretical background, implementation in simulation and experimental validation of two strategies for realising aperiodic control systems. The two strategies are event-triggered and self-triggered control. The Quanser 3 DOF hover, an experimental drone-like system, serves as the basis for both the simulation model and experimental validation of the two strategies. Event-triggered control is a reactive strategy, meaning it continuously monitors the state of the system to update the control action aperiodically. Whereas self-triggered control is a proactive strategy, meaning that the state of the system is sampled aperiodically, as well as the update of the control action. Both strategies are designed based on the solution of a linear matrix inequality derived from a Lyapunov function. It is found that both strategies perform well in simulation, but only the event-triggered control strategy provides adequate performance on the Quanser 3 DOF hover. The reason why the self-triggered control strategy performs inadequately when tested on the Quanser 3 DOF hover is likely due to unmodelled dynamics and disturbances.

Acknowledgements

I would like to thank my supervisor, Damiano Rotondo, for outstanding help, insights and guidance during the project! I also want to express my appreciation for the collaborative experience of organising a memorable Mario Kart tournament, which included students and employees of the university.

Contents

Abbreviations

BMI	**B**ilinear **M**atrix **I**nequality
CCW	**C**ounter-**C**lock**W**ise
CW	**C**lock**W**ise
DOF	**D**egrees **O**f **F**reedom
ETC	**E**vent-**T**riggered **C**ontrol
LMI	**L**inear **M**atrix **I**nequality
LTI	**L**inear **T**ime **I**nvariant
SS	**S**tate **S**pace
STC	**S**elf-**T**riggered **C**ontrol
UF	**U**pdate **F**requency

Chapter 1

Introduction

Most commonly, control systems are realised with a periodic sampling rate and periodic updates of the control action [1]. The advantage of periodic sampling is that if the time between samples is chosen to be sufficiently small, modelling of the control system can be done without having to make any significant assumptions about the sampling process. A drawback to periodic control is that updates of the control action and sampling of the state are conducted regardless of whether it is necessary. Hence, spending more energy, computation power and bandwidth to transfer the sampled data than necessary. For systems not restricted by these drawbacks, periodic control is possibly the best alternative to realise control systems due to its simplicity. On the other hand, for systems limited by any of these drawbacks, implementing techniques that only update the control action or sample the state when necessary will be beneficial in terms of saving resources. Only updating the control action or sampling the state when necessary is denoted as aperiodic control. The overall objective of this thesis is to investigate two techniques for implementing aperiodic control.

The first technique for realising aperiodic control is event-triggered control (ETC). The main idea behind the ETC strategy is to update the control action aperiodically and sample the system's state continuously. Updates of the control action are conducted when a designed triggering condition violates a specified threshold. There are different methods of designing triggering conditions, where the common factor is that the algorithm continuously monitors the system's state and compares it to the state of the system at the last time the control action was updated. In this thesis, the triggering condition of the controller is designed based on the derivative of a quadratic Lyapunov function.

The second technique for realising aperiodic control is self-triggered control (STC). Similar to ETC, STC updates the control action aperiodically. Additionally, STC samples the system's state aperiodically. The main idea behind STC is to plan the next instance at which sampling of the system's state and updating the control action is required. This is done based on a comparison between the predicted state trajectory of the system and an exponentially decreasing function based on the desired decay rate of the system. The predicted state trajectory is computed based on the sampled state, control input and a model of the system. The exponentially decreasing function is based on the sampled state of the system. The difference between ETC and STC is that ETC relies on continuous monitoring of the state, meaning that it is reactive, while STC relies on a simulated model and prediction of the state, meaning that it is proactive.

The two control strategies are implemented on the Quanser 3 degrees of freedom (DOF) hover [2], which can be seen in Figure 2.1. The control strategies are tested and verified in a simulation environment modelled based on the 3 DOF hover and on the actual 3 DOF hover in the laboratory of UiS. The 3 DOF hover is a stationary quadcopter with 3 DOF, yaw, pitch, and roll, which is configured to be controlled by *MATLAB* and *SIMULINK*. The experiments conducted in the thesis test the self-stabilising capabilities of the two control strategies. Additionally, the first presented ETC strategy is further developed to be able to follow a reference signal, which is tested in simulation and by experimental validation.

1.1 History

Most articles that discuss aperiodic control have been published within the last 20 years. This does not mean that it only has been discussed recently. The first examples of articles discussing aperiodic control were published in the 1960s. Some examples of articles published in the 1960s are [3], [4] and [5]. One of the reasons why aperiodically control has been researched extensively in the last 20 years is the publication of [6], which proved that aperiodic control could be very effective in control systems, in addition to the publication of [7], which proved that aperiodic control has the potential to save significant amounts of computing resources. Both these articles were published in 1999. The first article discussing STC [8] emerged in 2003. In 2007, [9] was published, which implemented the ETC strategy on state space (SS) systems. In 2012, [10] was published, which is probably the most referred to article within the research field of aperiodic control. [10] gives introductory insights to both ETC and STC and presents the main ideas behind the strategies.

Some of the main developments for aperiodic control over the past ten years have been focused on physical implementations and use cases of the already defined theory. Because this thesis focuses on implementing and developing algorithms for a drone-like system, similar work has been researched in preparation for the thesis. Some examples are [11] and [12], which utilise ETC together with an LQR controller to achieve stabilisation from different initial conditions and disturbances on a quadrotor drone. What separates [11] and [12] from work done in this thesis is that the gain matrix is computed by solving a linear matrix inequality (LMI) rather than using the LQR method, additionally reference signal tracking has been implemented in this thesis. Another article that implements ETC on a quadrotor drone is [13]. This is done using a non-linear controller, which separates it from the controller implemented in this thesis.

1.2 Project Description

The following project description in *cursive* is the original one presented to me when I applied to write the thesis.

Event-triggered and self-triggered control of the 3 DOF hover system

Event-triggered control and safe-triggered control are two control approaches that consist of two elements: i) a feedback controller that computes the control action, and a triggering mechanism the determines when the control action is to be computed again. Event-triggered control is reactive since the triggering condition is based on monitoring the current measurements and detecting when some performance index becomes degrades. On the other hand, self-triggered control is proactive since the next control update time is precomputed according to model-based predictions. Both approaches have the potential of saving time when compared to traditional periodic sampling approaches.

The goal of the project is to test these two control strategies on the 3 DOF Hover produced by Quanser which is available in the laboratory at UiS. This equipment consists of a planar round frame with four propellers mounted on a three degree of freedom pivot joint that enables the body to rotate about the different axes (roll, pitch, and yaw). The propellers generate a lift force that can be used to directly control the pitch and roll angles.

Activities and Objectives

- *Literature study and analysis of the state-of-the-art of event-triggered and self-triggered control*
- *Implementation and evaluation of event-triggered control in a simulation environment*
- *Implementation and evaluation of self-triggered control in a simulation environment*
- *Experimental validation of the designed control laws on the 3 DOF Hover system available at the laboratory at UiS*

1.3 Thesis Outline

The remaining outline of this thesis is structured as follows:

- **Chapter 2** presents the system description of the Quanser 3 DOF hover, which consists of mathematical modelling and SS representation of the system.
- **Chapter 3** contains the theoretical background on which the thesis is based. The Chapter starts with sections defining theoretical concepts used in the ETC and STC strategies. These sections are followed by longer sections that present the ETC and STC strategies.
- **Chapter 4** presents the experiments that have been conducted and the results of these in the form of figures, tables and some discussion.
- **Chapter 5** discusses the results presented in Chapter 4 to a greater extent. This Chapter also compares the strategies and gives some ideas for future work that can be of interest.
- **Chapter 6** contain the final conclusion of the thesis.

Chapter 2

System Description

The control strategies presented in the thesis are tested on a simulation model based on the 3 DOF hover, referred to as *simulation*, and by experimental validation on the physical Quanser 3 DOF hover depicted in Figure 2.1, referred to as the *3 DOF hover*. The design of the simulation model and the setup of the physical model are based on the works found in [14], published by the producer Quanser. The 3 DOF hover is a stationary quadcopter with 3 DOF, yaw, pitch, and roll, which is configured to be controlled by *MATLAB* and *SIMULINK* [14]. The 3 DOF hover consists of four propellers driven by individual DC motors, mounted on a planar frame which is coupled to the stationary base pedestal [14]. The angles of the three axes are controlled by the propellers and measured by high-resolution encoders. The transmitted and received signals, motor commands and encoder signals, respectively, are communicated through a slip ring mechanism, which allows the yaw axis to move continuously without limitations.

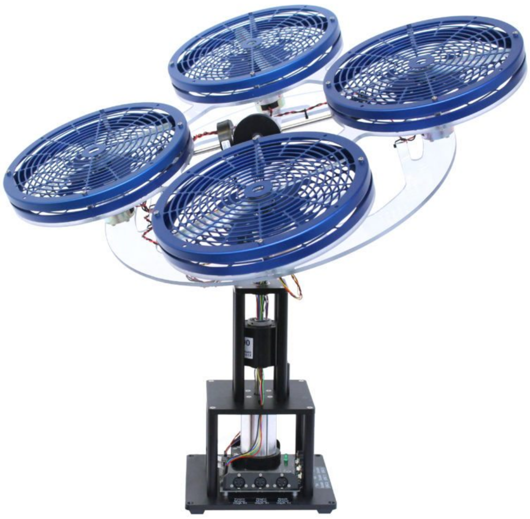


Figure 2.1: The Quanser 3 DOF hover [14].

2.1 Modeling

Figure 2.2 shows a free-body diagram of the physical system depicted in Figure 2.1. The positive rotation of the three axes is defined, as well as the positive direction of actuated force by the individual propellers on the system. The front and back propellers have been defined as having a positive rotation direction as counter-clockwise (CCW), and the right and left propellers have been defined as having a positive rotation direction as clockwise (CW).

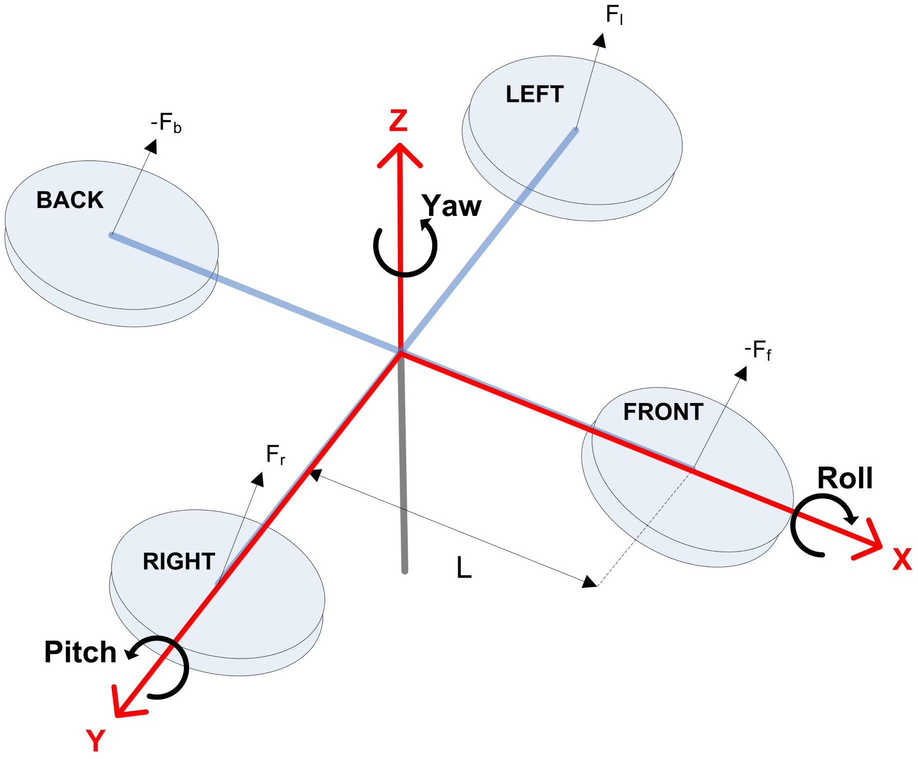


Figure 2.2: Free-body diagram of 3 DOF Quanser aero [14].

2.1.1 Pitch and Roll Axes

The dynamics for the roll and pitch axes can, in general, be denoted as:

$$J_i\ddot{\theta}_i(t) = \Delta F(t)L \tag{2.1}$$

where $i = p$ for pitch angle and $i = r$ for roll angle. J_i is used to describe the moment of inertia about the pitch/ roll axes, $\ddot{\theta}_i(t)$ is the double time derivative of the pitch/ roll angle relative to the pivot point, i.e. the acceleration about the pitch/ roll axes, $\Delta F(t)$ describes the differential thrust-force (difference between $F_f(t)$, $F_b(t)$, $F_r(t)$ and $F_l(t)$), actuated by the propellers and L is the distance from the pivot to the individual propellers. Figure 2.3 presents a free-body diagram for the pitch axis. The description also holds for the roll axis if relevant variables are changed.

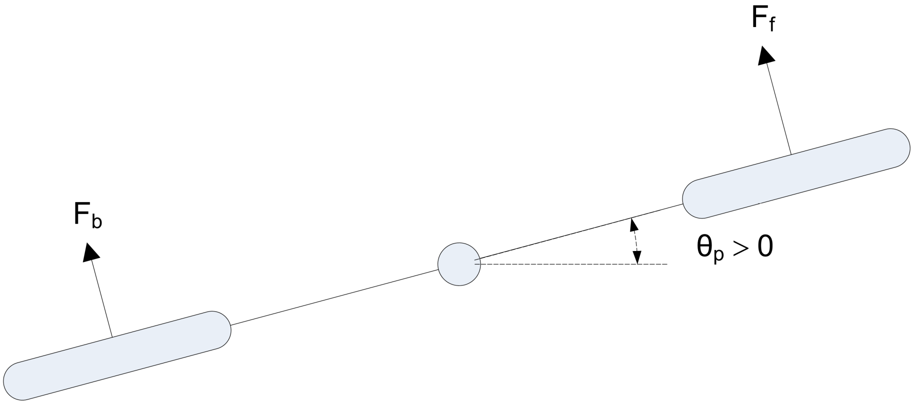


Figure 2.3: Free-body diagram of pitch axis [14]. This free-body diagram also holds for the roll axis if we change to the relevant variables.

To further extend the general expression in Equation (2.1) for the pitch axis, the DC motor voltages $V_f(t)$ and $V_b(t)$ and a force-thrust constant K_f of the motors/ propellers are introduced. The expression for the thrust forces can be expanded to $F_i(t) = K_f V_i(t)$. This results in the expression:

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

where the relevant symbols are described in Table 2.1.

Table 2.1: Parameters and variables related to the pitch axis.

Symbol	Description	Value	Unit
L	Distance from the pivot to the motor	0,197	m
J_p	Moment of inertia about the pitch axis	0,0552	kg m^2
K_f	Force-thrust constant of motor/ propeller	0,1188	N/ V
$\theta_p(t)$	Pitch angle	Variable	r
$V_f(t)$	Front motor voltage	Variable	V
$V_b(t)$	Back motor voltage	Variable	V

The dynamics of the roll axis can be derived in the same way as the pitch axis. This is because the axes are perpendicular to each other. The dynamics for the roll axis are:

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

where the relevant symbols are described in Table 2.2.

Table 2.2: Parameters and variables related to the roll axis.

Symbol	Description	Value	Unit
L	Distance from the pivot to the motor	0,197	m
J_r	Moment of inertia about the roll axis	0,0552	kg m^2
K_f	Force-thrust constant of motor/ propeller	0,1188	N/ V
$\theta_r(t)$	Pitch angle	Variable	r
$V_r(t)$	Right motor voltage	Variable	V
$V_l(t)$	Left motor voltage	Variable	V

2.1.2 Yaw Axis

The dynamics of the yaw axis can be described by asserting all torques actuated by the rotating propellers:

$$J_y\ddot{\theta_y}(t) = \Delta\tau(t) = \tau_l(t) + \tau_r(t) - \tau_f(t) - \tau_b(t) \tag{2.4}$$

where $\Delta\tau(t)$ is the difference in torque exerted on the system by all the individual motors. Figure 2.4 presents a free-body diagram describing the yaw axis's movement and dynamics.

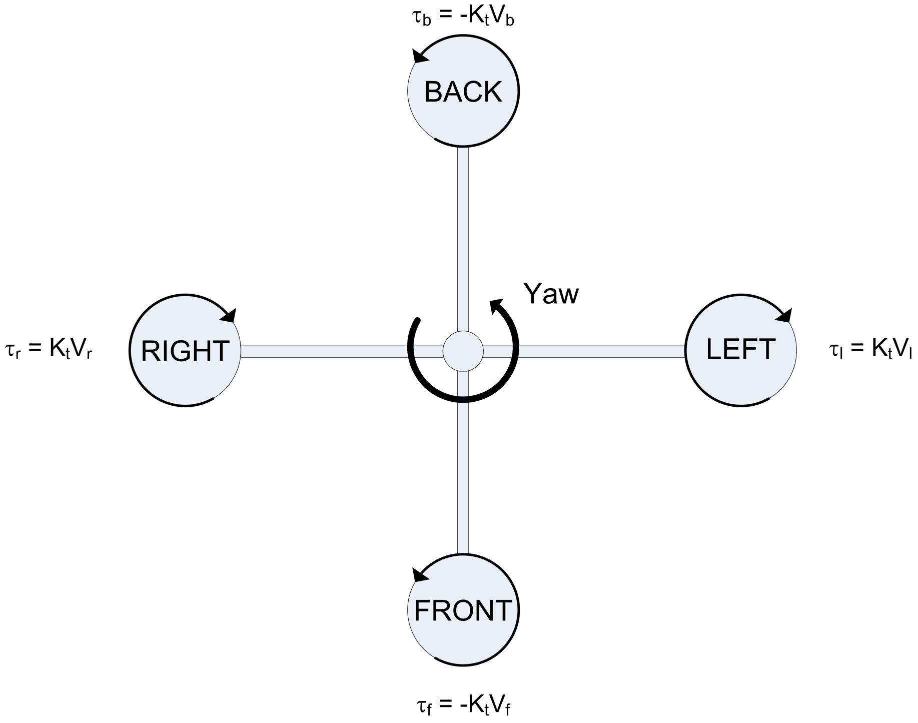


Figure 2.4: Free-body diagram of yaw axis [14].

A more specific term can be derived for the dynamics of the yaw axis in the same way as it was done for the roll and pitch axes. The torques actuated by the propellers are replaced by the motor voltages multiplied by the force-thrust constant of the motors, $\tau_i(t) = K_t V_i(t)$. Resulting in the extended-expression:

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

where the relevant symbols are described in Table 2.3.

Table 2.3: Parameters and variables related to the yaw axis.

Symbol	Description	Value	Unit
J_y	Moment of inertia about the yaw axis	0,110	kg/ m^2
K_t	Torque thrust constant of motor/ propeller	0,0036	Nm/ V
$\theta_y(t)$	Yaw angle	Variable	r
$\tau_l(t)$	Torque generated by left CW rotor	Variable	Nm
$\tau_r(t)$	Torque generated by right CW rotor	Variable	Nm
$\tau_f(t)$	Torque generated by front CCW rotor	Variable	Nm
$\tau_b(t)$	Torque generated by back CCW rotor	Variable	Nm

2.2 State-space Model

SS models are mathematical representations used to describe the behaviour of a dynamical system in a compact form. The simplest SS representation is the linear SS model, which is the form in which the 3 DOF hover is represented in this thesis. In general, SS models for linear time invariant (LTI) systems can be written on the form [15]:

$$\begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{cases} \tag{2.6}$$

where the relevant symbols are described in Table 2.4. The state vector describes the current state of the system. For the 3 DOF hover, an appropriate choice for the state vector is for it to contain the angular position and the angular velocity for the three axes, resulting in the state vector:

$$x^T(t) = [\theta_y(t) \quad \theta_p(t) \quad \theta_r(t) \quad \dot{\theta}_y(t) \quad \dot{\theta}_p(t) \quad \dot{\theta}_r(t)] \tag{2.7}$$

Together with the output vector containing the angular positions:

$$y^T(t) = [\theta_y(t) \quad \theta_p(t) \quad \theta_r(t)] \tag{2.8}$$

Table 2.4: Notations related to the SS model. The dimensions of the matrices and vectors in the table are defined by $n-$ states, $m-$ inputs and $p-$ outputs.

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

The input vector for the system is described by what act on the system, which for the 3 DOF hover is the voltages of the individual motors:

$$u^T(t) = [V_f(t) \quad V_b(t) \quad V_r(t) \quad V_l(t)] \tag{2.9}$$

The state matrix is derived by considering the relation between $\dot{x}$ and x, which for the 3 DOF hover results in:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.10}$$

To formulate the input matrix, B, Equations (2.2) - (2.4) are considered, together with the control vector , given by Equation (2.9), leading to:

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} \\ \frac{LK_f}{J_p} & -\frac{LK_f}{J_p} & 0 & 0 \\ 0 & 0 & \frac{LK_f}{J_r} & -\frac{LK_f}{J_r} \end{bmatrix} \tag{2.11}$$

The output matrix, y, is defined by considering the output vector denoted in Equation (2.8) and the state vector denoted in Equation(2.7):

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{2.12}$$

Finally, for the SS model, the feedthrough matrix, D, is a null matrix:

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.13}$$

When the SS system given by Equation (2.6) is discussed going forward in this thesis, the term containing the feedthrough matrix will be left out, as all the elements of the matrix are zero. By combining all the derived matrices denoted in Table 2.4 into the expression for the SS system, given by Equation (2.6), the compact SS representation for the 3 DOF hover is defined as:

$$\begin{cases} \begin{bmatrix} \dot{\theta}_y(t) \\ \dot{\theta}_p(t) \\ \dot{\theta}_r(t) \\ \ddot{\theta}_y(t) \\ \ddot{\theta}_p(t) \\ \ddot{\theta}_r(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_y(t) \\ \theta_p(t) \\ \theta_r(t) \\ \dot{\theta}_y(t) \\ \dot{\theta}_p(t) \\ \dot{\theta}_r(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} \\ \frac{LK_f}{J_p} & -\frac{LK_f}{J_p} & 0 & 0 \\ 0 & 0 & \frac{LK_f}{J_r} & -\frac{LK_f}{J_r} \end{bmatrix} \begin{bmatrix} V_f(t) \\ V_b(t) \\ V_r(t) \\ V_l(t) \end{bmatrix} \\ \\ \begin{bmatrix} \theta_y(t) \\ \theta_p(t) \\ \theta_r(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_y(t) \\ \theta_p(t) \\ \theta_r(t) \\ \dot{\theta}_y(t) \\ \dot{\theta}_p(t) \\ \dot{\theta}_r(t) \end{bmatrix} \end{cases} \tag{2.14}$$

Chapter 3

Theoretical Background

3.1 Linear Quadratic Regulator

A popular control strategy for SS systems is the linear quadratic regulator (LQR), which utilises a feedback loop. LQR is a feedback control method based on minimising a cost function. The general form of the cost function, used for the SS system, representing the 3 DOF hover in this case, consists of quadratic terms of the state vector $x(t)$ and the input vector $u(t)$ on the form [16]:

$$J = \int_0^\infty (x^T(t)Qx(t) + u^T(t)Ru(t))dt \tag{3.1}$$

where the matrices Q and R are weighing matrices specified by design. Q and R are positive definite and symmetric matrices. Where a symmetric matrix is defined as a quadratic matrix that is symmetric about its diagonal, this can formally be written as $B = B^T$, where B is an arbitrary quadratic and symmetric matrix. A matrix is defined as a positive definite matrix if all the eigenvalues of a quadratic matrix are real and positive. It can be proven that the optimal control law for the LQR strategy is [16]:

$$u(t) = -R^{-1}B^TP(t)x(t) = -Kx(t) \tag{3.2}$$

where $P(t)$ is found by solving a Riccati differential equation of the form[16]:

$$A^TP(t) + P(t)A - P(t)BR^{-1}B^TP(t) + Q = 0 \tag{3.3}$$

Appendix B, Figure B.1 presents an example of a system controlled by an LQR controller in a feedback loop. This example is conducted to compare with the performance of one of the experiments conducted in Chapter 4. Figure 3.1 is an example of a typical periodic feedback control system in which the LQR strategy could be applied.

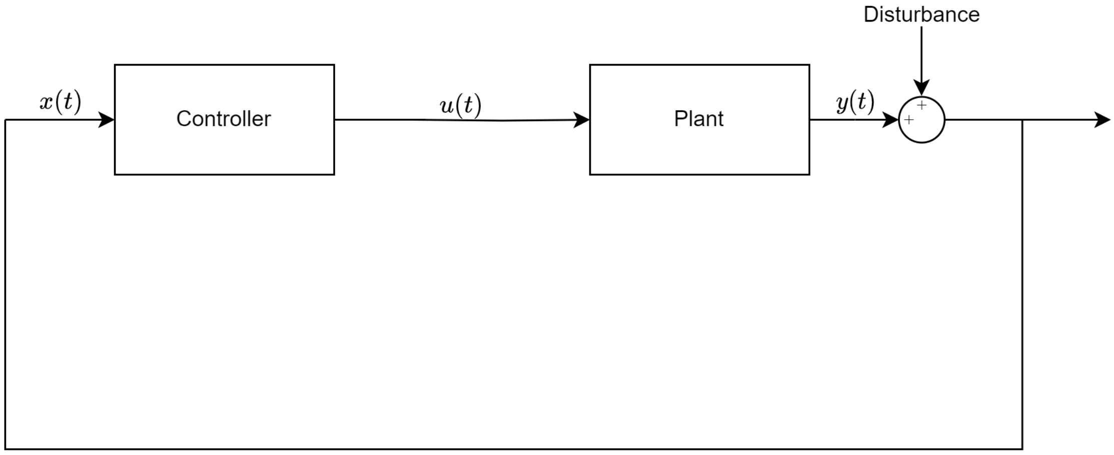


Figure 3.1: Block diagram of a feedback loop.

3.2 State-space Trajectories

One of the main components of the STC algorithm, which is presented later in Section 3.5, is the computation of the expected state trajectory of the system. The state trajectory is the first equation in the equation set of the SS equations. The SS trajectory is often referred to as the solution to the SS equations. The general form of the SS equations is:

$$\begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{cases} \tag{3.4}$$

The solution/ trajectory of the SS equations is [17]:

$$\begin{cases} x(t) = e^{A(t-t_0)}x(t_0) + \int_0^t e^{A(t-s)}Bu(s)ds \\ y(t) = Ce^{A(t-t_0)}x(t_0) + C\int_0^t e^{A(t-s)}Bu(s)d \end{cases} \tag{3.5}$$

where t_0 is initial time and $x(t_0)$ is the initial state. The state trajectory presented in Equation (3.5) can be used to compute the state at the time, t, in Figure 3.2 the output trajectory of the 3 DOF hover described by Equation (2.14) is shown. Only states the output trajectory consisting of states x_1, x_2 and x_3 are shown in the figure. The states remaining states of the state vector, x_4, x_5 and x_6, which are the angular velocity of the system, are not shown here since it is easier to get a visual understanding of the angular positions rather than the angular velocity. Moreover, only the output, $y(t)$, consisting of the angular position of the three axes, is presented in the figures throughout the thesis, even though it is the full state that is used in computations.

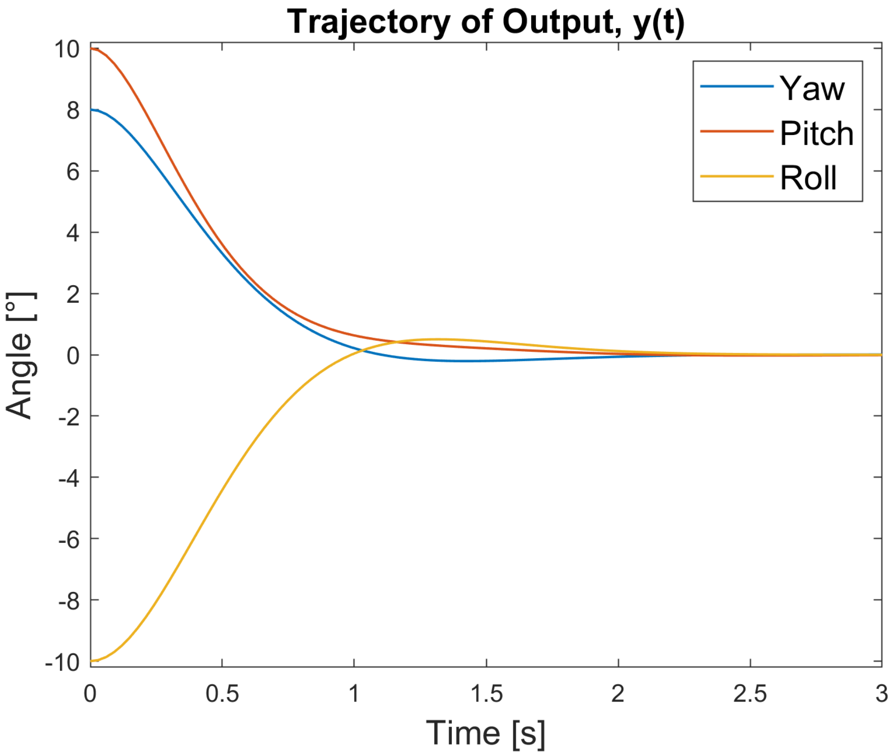


Figure 3.2: Example of SS trajectory. The system is initialised from a state where the yaw, pitch and roll axes are 8, 10 and -10 degrees, respectively.

The system is controlled by a feedback controller $u(t) = Kx(t)$. Later, in Section 3.4, it is described how the gain matrix, K, is computed.

3.2.1 Tustin Discretisation

One challenge when implementing the state trajectory of Equation (3.5) in the STC algorithm is the computation of the integral. This is because integrals generally are functions of continuous time, whereas when implemented on digital platforms, the integrals have to be implemented in discrete time. There are several methods for discretising a signal. Some examples are forward Euler discretisation, backward Euler discretisation, Tustin discretisation, and zero-order hold discretisation. In this thesis, Tustin discretisation is the preferred method for discretisation because it preserves stability, it is relatively easy to implement, and it produces an approximation with low error [18]. Whereas the two Euler methods are also easy to implement but are less accurate compared to the Tustin method, forward Euler does not guarantee that stability is preserved, and zero-order hold is more complex to implement.

Tustin approximation of an integral can be computed as[19]:

$$
\begin{aligned}
y(t_{n+1}) &= y(t_n) + \int_{t_n}^{t_{n+1}} f(t,y(t))dt \\
&\approx y(t_n) + \frac{h}{2}[f(t_n, y(t_n)) + f(t_{n+1}, y(t_{n+1}))]
\end{aligned}
\tag{3.6}
$$

where $h = t_{n+1} - t_n$. Visually, Equation (3.6) can be thought of as the total area contained by the red lines, which is the Tustin approximate of the area below the function defined by the blue line.

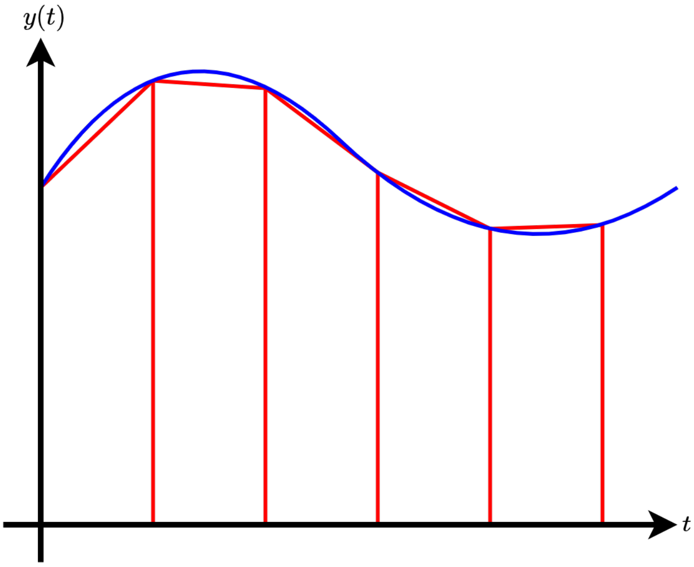


Figure 3.3: The area under the function, represented by the blue line, can be approximated using the Tustin approximation, represented by the area contained by the red lines.

3.3 Lyapunov Functions

Lyapunov functions are commonly used to analyse dynamical systems to determine a system's stability. A Lyapunov function is a scalar function, which is defined to be decreasing, i.e. the derivative of the Lyapunov is, by definition, negative [20]. Lyapunov functions can be thought of as a scalar function which provides information about the energy contained in the current state of the system. A commonly utilised Lyapunov function for LTI SS representation is the quadratic Lyapunov function. Consider the SS system defined in Equation (3.4), then the quadratic Lyapunov function of the system is:

$$
V(x(t)) = x(t)^T P x(t) \tag{3.7}
$$

where P is a positive definite and symmetric matrix. The following Subsection 3.3.1 explains more about how the matrix P is computed. This thesis utilises quadratic Lyapunov functions in both the ETC and STC strategies. Figure 3.4 presents the quadratic Lyapunov function of the previously provided example depicted in Figure 3.2.

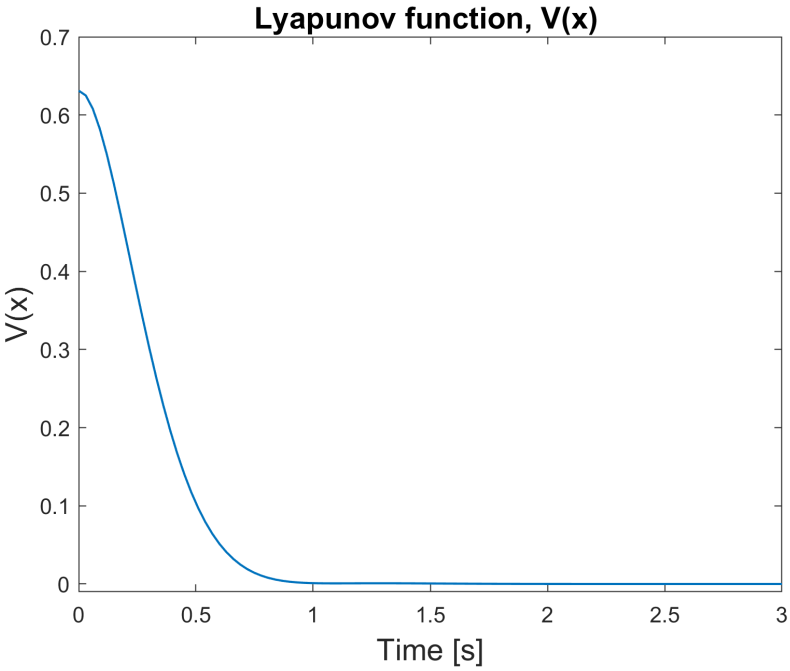


Figure 3.4: Quadratic Lyapunov function of the SS trajectory shown in Figure 3.2.

3.3.1 Linear Matrix Inequalities and the Derivative of the Lyapunov Function

LMIs refer to matrix inequalities consisting of matrix variables that are linear, as the name implies. Later in this Chapter, when introducing the ETC and STC strategies LMIs are encountered when deriving the respective algorithms. To lay the groundwork for the later presented control strategies, this section provides a simple example of how a quadratic Lyapunov function is used to derive a general LMI problem. First, the following SS system is considered:

$$\dot{x}(t) = Ax(t) \tag{3.8}$$

where the quadratic Lyapunov function of the system is:

$$V(x(t)) = x^T(t)Px(t) \tag{3.9}$$

where P is a positive definite and symmetric matrix. The derivative of the Lyapunov function in the Equation (3.9) is:

$$\begin{aligned}\dot{V}(x(t)) &= \dot{x}^T(t)Px(t) + x^T(t)P\dot{x} \leq 0\\ &= x^T(t)A^TPx(t) + x^T(t)PAx(t) \leq 0\\ &= x^T(t)\underbrace{\Big(A^TP + PA\Big)}_{LMI}x(t) \leq 0 \end{aligned} \tag{3.10}$$

where the LMI is marked by the curly bracket in the last line of the equation. This LMI is on the form most commonly encountered in the field of control theory [20].

$$A^TP + PA \leq 0 \tag{3.11}$$

The LMIs are solved using the toolbox *YALMIP* [21] together with the solver *Mosek* in *MATLAB*. One example of how an LMI like the one in Equation (3.11) can be solved using *YALMIP* is shown in Listing 3.1:

```
yalmip('clear');
P = sdpvar(6);
LMI = A'*P + P*A;
tol = 1e-10;
value = tol*eye(6);

Constraints = [P >= value, LMI <= -value];
ops = sdpsettings('solver', 'mosek', 'verbose', 0);
res_opt = optimize(Constraints, [], ops);

P = double(P);
```

Listing 3.1: Using *YALMIP* to solve LMI denoted in Equation (3.11).

3.4 Event-triggered Control

The main purpose of the ETC strategy is to reduce the number of times the controller updates the control action, i.e. saving resources in the control system. This is achieved by updating the control action aperiodically instead of periodically. To implement ETC, a triggering condition is required. When the value of the triggering condition is greater than a specified threshold, the control action is updated.

There are several different ways in which a triggering condition can be developed. The simplest implementation of a triggering condition is based on comparing the measured state of the system with the state of the system at the previous time of which the control action was updated and checking if the difference between the measurements is below a specified performance threshold, an algorithm for this type of ETC is presented in [7]. Another way of implementing ETC is by designing a triggering condition based on a Lyapunov function. Similar to the first mentioned implementation, the Lyapunov-based implementation measures the state of the system and the state of the system at the previous time at which the control action was updated, but instead of comparing these two measurements directly and checking a specified threshold, it considers a quadratic Lyapunov function with the two measurements as input variables and checks if the decay rate/ derivative of the Lyapunov function is inline with a decay rate specified by design. In this thesis, an algorithm based on the Lyapunov function is used to implement ETC for the 3 DOF hover, both in a simulation environment and on the 3 DOF hover. The ETC algorithm developed and presented here is based on the work found in [9], [10] and [22]. Figure 3.5 presents the main idea behind ETC visually, implemented in a feedback loop, as a block diagram. The figure highlights that the triggering condition uses the measured state of the system and the state of the system at the previous time at which the control action was updated as input arguments.

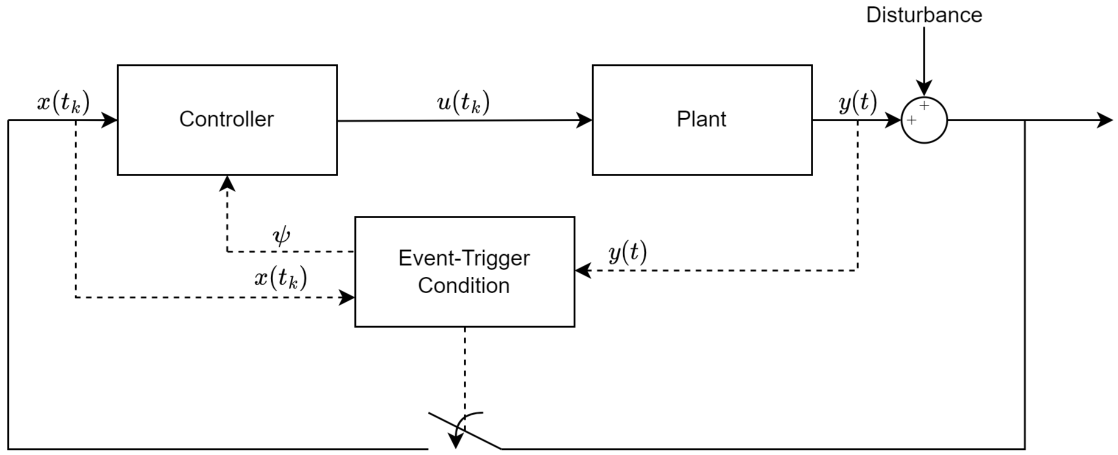


Figure 3.5: Block diagram visually presenting the main idea behind ETC.

In order to introduce ETC, the following SS system is considered:

$$\begin{cases} \dot{x}(t) & = Ax(t) + Bu(t) \\ y(t) & = Cx(t) \end{cases} \tag{3.12}$$

and the linear feedback control law:

$$u(t) = Kx(t) \tag{3.13}$$

By combining the first equation of the SS system given by Equation (3.12) with the linear feedback control law given by Equation (3.13), the following closed-loop system is obtained:

$$\dot{x}(t) = (A + BK)x(t) \tag{3.14}$$

The closed-loop system denoted in Equation (3.14), is used when developing the ETC algorithm. Next, the following quadratic Lyapunov function is defined:

$$V(x(t)) = x^T(t)Px(t) \tag{3.15}$$

where P is a symmetric and positive definite matrix. Furthermore, to design a triggering condition for the ETC algorithm, the derivative of the Lyapunov function, denoted in Equation (3.15) is defined, based on the closed-loop system presented in Equation (3.14):

$$\begin{aligned}\dot{V}(x(t)) &= \dot{x}^T(t)Px(t) + x^T(t)P\dot{x}(t)\\ &= x^T(t)(A + BK)^T Px(t) + x^T(t)P(A + BK)x(t)\\ &= x^T(t)\underbrace{[(A + BK)^T P + P(A + BK)]}_{-Q}x(t)\\ &= -x^T(t)Qx(t)\end{aligned} \tag{3.16}$$

where Q is the specified rate of decrease for the derivative of the Lyapunov function, $\dot{V}(x(t))$. Since the derivative of Lyapunov functions, as discussed in Section 3.3, is negative, $\dot{V}(x(t))$ is always negative. In order to formulate an inequality which can be used to design the triggering condition for the controller, $\dot{V}(x(t))$ is required to be less or equal to the specified rate of decrease, $-x^T(t)Qx(t)$:

$$\dot{V}(x(t)) \leq -x^T(t)Qx(t) \tag{3.17}$$

In some cases, it may be desired to specify a slower rate of decrease than what is defined in Equation (3.17). This can be achieved by introducing a parameter $\sigma \in [0,1)$, which leads to the weaker inequality [10]:

$$\dot{V}(x(t)) \leq -\sigma x^T(t)Qx(t) \tag{3.18}$$

which is a weaker inequality than the one presented in Equation (3.17), meaning it will have a slower rate of decrease. Smaller values of σ lead to a slower rate of decrease, larger values of σ will lead to faster rates of decrease and $\sigma = 1$ yield the same inequality as denoted in Equation (3.17). In Appendix C, the effect of different values of σ can be seen. Later in Chapter 4, the various plots in Appendix C are explained to a greater extent. After introducing σ, $\dot{V}(x(t))$ can no longer be considered a Lyapunov function because Lyapunov functions are required to be decreasing. Instead, $\dot{V}(x(t))$ can be called a Lyapunov-like function, but for simplicity in the context of this thesis, $\dot{V}(x(t))$ will still be referred to as a Lyapunov function.

In order to calculate the Lyapunov function given by Equation (3.15) and the derivative of the Lyapunov function given by Equation (3.16), the matrices K, P and Q has to be defined. The matrices are found by solving the bilinear matrix inequality (BMI), which is a factor in the derivative of the Lyapunov function given by Equation (3.16):

$$Q = -[(A + BK)^T P + P(A + BK)] \leq 0, \tag{3.19}$$

this is a BMI due to the product between variables P and K, which both are matrices that are defined by the solution of the BMI. The BMI is a non-convex optimisation problem which generally is hard to solve. A better approach to solve the inequality is to redefine the BMI presented in Equation (3.19) to an LMI by introducing the transformation $Y = KP^{-1}$ and apply the Schur complement [23]. LMIs are convex optimisation problems, which in comparison to BMIs, are easy to find a solution to. By doing the substitution and applying the Schur complement, the following expression is derived:

$$\begin{aligned} Q &= -[(A + BK)^T P + P(A + BK)] \leq 0 \\ &= -[A^T P + K^T B^T P + PA + PBK] \leq 0 \\ &= -[P^{-1} A^T + P^{-1} K^T B^T + AP^{-1} + BKP^{-1}] \leq 0 \\ &= -[AP^T + BY + PA^T + Y^T B^T] \leq 0 \end{aligned} \tag{3.20}$$

which can be solved using the same methodology as presented in Subsection 3.3.1, using *YALMIP* [21]. The specific implementation in *MATLAB* used to find the solution of the LMI denoted in Equation (3.20) is shown in Appendix E.1. The solution of the LMI is defined by the matrices Y and P, which can be used to compute $K = YP$ and $Q = -[AP^T + BY + PA^T + Y^T B^T]$. When designing the triggering condition, it has to be considered that the linear feedback control law, presented in Equation (3.13), is kept constant between sampling and update instances, which is denoted as:

$$u(t) = u(t_k) = Kx(t_k) \tag{3.21}$$

As a consequence of the updated linear feedback control law introduced in Equation (3.21), the closed-loop system, presented in Equation (3.14), is changed to:

$$\dot{x}(t) = Ax(t) + BKx(t_k) \tag{3.22}$$

As discussed earlier, the triggering condition takes the measured state of the system and the state of the system at the previous time of which the control action as input variables, this is formulated as:

$$\Delta x(t) = x(t_k) - x(t) \tag{3.23}$$

which substituted into Equation (3.22) yields the following closed-loop system:

$$\dot{x}(t) = (A + BK)x(t) + BK\Delta x(t) \tag{3.24}$$

Looking back to the derivative of the Lyapunov function defined in Equation (3.16), all matrices have now been defined, and $\dot{x}^T(t)$ has to be derived to find the final expression. The transposed of the state derivative can be formulated as:

$$\begin{aligned}\dot{x}^T(t) &= [(A + BK)x(t) + BK\Delta x(t)]^T\\ &= [(A + BK)x(t)]^T + [BK\Delta x(t)]^T\\ &= x^T(t)(A + BK)^T + (BK\Delta x(t))^T\end{aligned} \tag{3.25}$$

Furthermore, the derivative of the Lyapunov function for the new closed-loop system is found by substituting Equations (3.24) and (3.25) in Equation (3.16), as shown below.

$$\begin{aligned}\dot{V}(x(t)) &= \dot{x}^T(t)Px(t) + x^T(t)P\dot{x}(t)\\ &= [x^T(t)(A + BK)^T + (BK\Delta x(t))^T]Px(t)\\ &+ x^T(t)P[(A + BK)x(t) + BK\Delta x(t)]\\ &= x^T(t)(A + BK)^T Px(t) + x^T(t)P(A + BK)x(t) + (BK\Delta x(t))^T Px(t)\\ &+ x^T(t)PBK\Delta x\\ &= x^T(t)[(A + BK)^T P + P(A + BK)]x(t) + 2x^T(t)PBK\Delta x(t)\\ &= -x^T(t)Qx(t) + 2x^T(t)PBK\Delta x(t)\end{aligned} \tag{3.26}$$

As discussed when Equation (3.17) was presented, the variable $\sigma \in [0,1)$ is introduced to be able to specify a slower rate of decrease. This leads to the following derivative of the Lyapunov function:

$$\dot{V}(x(t)) = -x^T(t)Qx(t) + 2x^T(t)PBK\Delta x(t) \leq -\sigma x^T(t)Qx(t) \tag{3.27}$$

which is defined as the triggering condition:

$$\psi = (\sigma - 1)x^T(t)Qx(t) + 2x^T(t)PBK\Delta x(t) \leq 0 \tag{3.28}$$

ψ is introduced to represent the trigger condition. In Chapter 4, containing the experiments of this thesis, ψ is used when discussing the trigger condition. The implementation of the triggering condition, ψ, in *MATLAB* can be seen in Appendix E.3.

3.4.1 An Improved Gain Matrix

During testing and experimentation, it was found that the initial gain matrix, K, which is found by solving the LMI in Equation (3.20) and substituting into $K = YP$, is not very effective. This is because the elements of the derived gain matrix are relatively small, which in turn, results in conservative error compensation in the feedback loop. One method which can be utilised to find a more effective/aggressive gain matrix is to move the poles of the closed-loop system further into the left half plane, as illustrated in Figure 3.6.

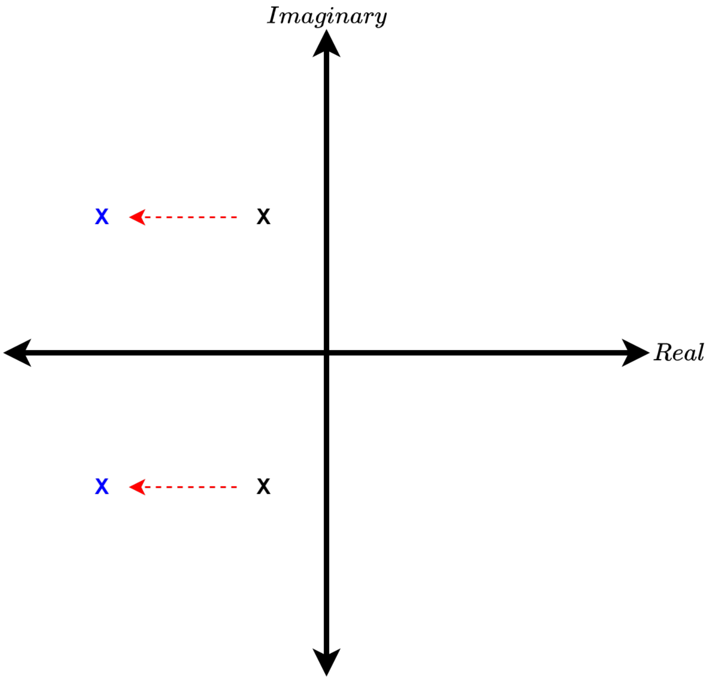


Figure 3.6: Conceptual graph shoving movement of poles further into the left half plane.

Definition 4.1: *" Let $\mathbb{D}$ be a domain on the complex plane, which is symmetric about the real axis. Then, a matrix $A \in \mathbb{R}^{n \times n}$ is said to be $\mathbb{D}$-stable if " [20]*

$$\lambda_i(A) \in \mathbb{D}, \ i = 1, 2, ..., n. \ [20] \tag{3.29}$$

The idea of $\mathbb{D}$-stability has to be considered to obtain a gain matrix that yields faster and more precise control of the system. The idea is to establish LMI conditions for the $\mathbb{D}$-stability of a matrix with different given $\mathbb{D}$-regions [20]. In other words, this means that conditions are introduced when solving the LMI, which moves the stability region of the system further into the left-half plane. In practice, this means forcing the system's closed-loop eigenvalues further into the left half-plane. This can be written formally as presented in Equation (3.30) and observed graphically in Figure 3.7.

$$\mathbb{D} = \mathbb{H}_\alpha = \{x + jy \mid x < -\alpha\} \tag{3.30}$$

where $\mathbb{H}_\alpha$ is a Hurwitz stable set. Hurwitz stability, $\mathbb{H}$, is commonly used to check if a system is stable. A system is asymptotically stable, according to the Hurwitz criterion, if the real part of all the eigenvalues of the system is strictly negative [24]. This is formally written as $Re[\lambda_i(A + BK)] < 0$. This means that the closed-loop eigenvalues of the system are contained in the shaded area/ $\mathbb{D}$-region of Figure 3.7, and thus, the system is stable.

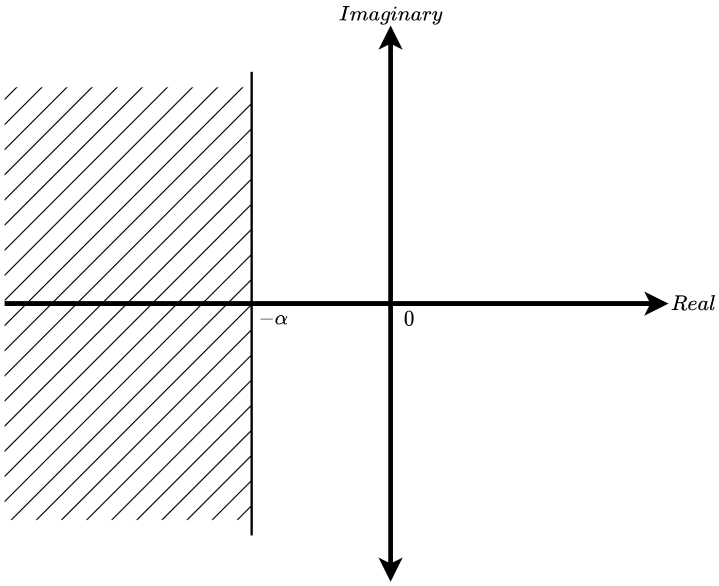


Figure 3.7: Conceptual graph, showing $\mathbb{D}$-region being moved further into the left half-plane.

To alter the $\mathbb{D}$-region, a term containing a constraint is introduced to the derived LMI. The new LMI with the added constraint is shown in Equation (3.31), which requires that all the poles of the system are located within the specified $\mathbb{D}$-region.

$$AP^T + BY + PA^T + Y^T B^T \leq -2\alpha P \text{ [20]} \tag{3.31}$$

where P is symmetric and positive definite and $K = YP$. In other words, the $\mathbb{D}$-region of the system is moved further into the left-half plane, depending on the chosen value of α. which can be solved using the same methodology as presented in Subsection 3.3.1, using *YALMIP*. The specific implementation in *MATLAB* used to find the solution of the LMI denoted in Equation (3.31) is shown in Appendix E.2.

3.4.2 Reference Tracking

The ETC strategy presented up to this point is designed to drive the system towards zero degrees in all axes, i.e. self-stabilise. However, for the 3 DOF hover and drone systems in general, it is desirable to be able to manoeuvre the system. To achieve manoeuvring of the system, reference tracking has to be introduced.

In general, tilting the pitch and roll axes of a drone system results in movements in the horizontal plane. The dynamics of the pitch and roll axes have previously been introduced in Equations (2.2) and (2.3). Similarly, the rotation of a drone can be achieved by controlling the motors as described in Equation (2.5). Reference tracking allows for specifying the angle of the three axes, allowing a drone to fly and manoeuvre as desired over distances.

A block diagram of the main idea behind reference tracking in ETC is shown in Figure 3.8. The purpose of introducing reference tracking is that it is desired that the system is able to follow a given reference signal, often different from zero. The ETC reference tracking algorithm developed and presented here is based on the works found in [25].

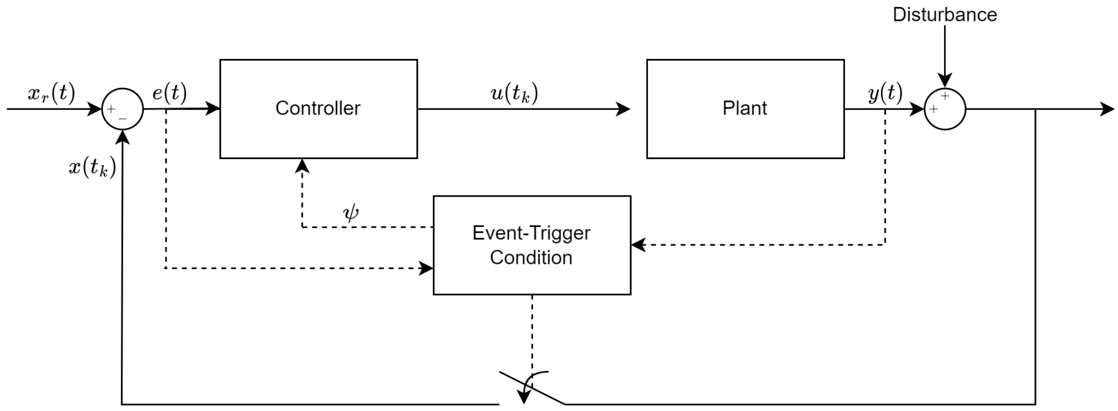


Figure 3.8: Block diagram visually presenting the main idea behind ETC reference tracking.

The triggering condition, presented in Equation (3.28) must be modified to allow the system to follow a reference signal. This is achieved by introducing a virtual system that generates a reference signal in addition to the previously introduced system description. Consider the previously presented SS system:

$$\begin{cases} \dot{x}(t) & = Ax(t) + Bu(t) \\ y(t) & = Cx(t) \end{cases} \tag{3.32}$$

and the sampled version of the linear feedback control law:

$$u(t_k) = Kx(t_k) \tag{3.33}$$

together with the defined expression for the difference between the measured state of the system and the state of the system at the previous time at which the control action was updated:

$$\Delta x(t) = x(t_k) - x(t) \tag{3.34}$$

By substituting Equation (3.34) into Equation (3.33), followed by Equation (3.33) into Equation (3.32), the following closed-loop system is:

$$\dot{x}(t) = (A + BK)x(t) + BK\Delta x(t) \tag{3.35}$$

Next, a system that generates the reference signal is introduced:

$$\begin{cases} \dot{x}_r(t) & = Ax_r(t) + Bu_r(t) \\ y_r(t) & = Cx_r(t) \end{cases} \tag{3.36}$$

and the sampled version of the linear feedback control law:

$$u_r(t_k) = Kx_r(t_k) \tag{3.37}$$

together with the defined expression for the difference between the measured state of the system and the state of the system at the previous time at which the control action was updated:

$$\Delta x_r(t) = x_r(t_k) - x_r(t) \tag{3.38}$$

By substituting Equation (3.38) into Equation (3.37), followed by Equation (3.37) into Equation (3.36), the following closed-loop system for the reference system is:

$$\dot{x}_r(t) = (A + BK)x_r(t) + BK\Delta x_r(t) \tag{3.39}$$

To find the final system description for the reference tracking problem, the system description, presented in Equation (3.35) and the reference system, presented in Equation

(3.39) are combined, resulting in the following expression:

$$\begin{aligned}\dot{e}(t) &= \dot{x}(t) - \dot{x}_r(t)\\ &= (A+BK)e(t) + BK\Delta x(t) - BK\Delta x_r(t)\end{aligned} \tag{3.40}$$

where $e(t) = x(t) - x_r(t)$. Furthermore, to obtain a Lyapunov function for the new system defined by Equation (3.40), the same steps presented previously are now considered with a more complex system. The quadratic Lyapunov function for the system is:

$$V(e(t)) = e(t)^T P e(t) \tag{3.41}$$

where P is symmetric and positive definite. Next, to define the triggering condition, the derivative of the Lyapunov function, defined in Equation (3.41) has to be defined, based on the closed-loop systems, presented in Equation (3.35) and the control law, presented in Equation (3.39). Before deriving the full expression, the expression for $\dot{e}^T(t)$ is considered:

$$\begin{aligned}\dot{e}^T(t) &= [(A+BK)e(t) + BK\Delta x(t) - BK\Delta x_r(t)]^T\\ &= e(t)^T(A+BK)^T + (BK\Delta x(t))^T - (BK\Delta x_r(t))^T\end{aligned} \tag{3.42}$$

By substituting the defined expressions, the derivative of the Lyapunov function presented in Equation (3.41) is:

$$\begin{aligned}\dot{V}(e(t)) &= \dot{e}(t)^T P e(t) + e(t)^T P \dot{e}(t)\\ &= [e(t)^T(A+BK)^T + (BK\Delta x(t))^T - (BK\Delta x_r(t))^T]Pe(t)\\ &+ e(t)^T P[(A+BK)e(t) + BK\Delta x(t) - BK\Delta x_r(t)]\\ &= e(t)^T[(A+BK)^T P + P(A+BK)]e(t)\\ &+ [(BK\Delta x(t))^T - (BK\Delta x_r(t)^T)]Pe(t)\\ &+ e(t)^T P[BK\Delta x(t) - BK\Delta x_r(t)]\\ &= -e(t)^T Q e(t)\\ &+ (BK\Delta x(t))^T Pe(t) + e(t)^T P(BK\Delta x(t))\\ &- (BK\Delta x_r(t)^T Pe(t) - e(t)^T P(BK\Delta x_r(t))\\ &= -e(t)^T Q e(t) + 2e(t)^T PBK\Delta x(t) - 2e(t)^T PBK\Delta x_r(t)\\ &= -e(t)^T Q e(t) + 2e(t)^T PBK(\Delta x(t) - \Delta x_r(t))\\ &= -e(t)^T Q e(t) + 2e(t)^T PBK\Delta e(t)\end{aligned} \tag{3.43}$$

As before, $\sigma \in [0,1)$ is introduced to specify a slower rate of decrease. This yields the following inequality:

$$\dot{V}(e(t)) = -e(t)^T Qe(t) + 2e(t)^T PBK\Delta e(t)\ leq - \sigma x^T(t)Qx(t) \tag{3.44}$$

which is defined as the triggering condition:

$$\psi = (\sigma - 1)e(t)^T Qe(t) + 2e(t)^T PBK\Delta e(t) \tag{3.45}$$

where ψ represents the triggering condition for the reference tracking problem. The implementation of the triggering condition, ψ, in *MATLAB* can be seen in Appendix E.4.

3.5 Self-triggered Control

The primary goal of STC, like ETC, is to reduce the number of times the controller has to update control action. Additionally, STC also saves resources by only sampling the state of the system when the control action is updated. This will save resources in the form of using the sensors of the system less, using less bandwidth, transferring the signal from the sensor and processing the signal. In the STC strategy, the controller decides when it is necessary to sample the state and update the control action. The controller decides this based on knowledge of the system's dynamics and the system's current sampled state. The main idea behind STC is to plan the next instance at which sampling and updating are required based on a comparison between the predicted state trajectory of the system, which is estimated based on the sampled state, control input and the model of the system, and an exponentially decreasing function, with a specified rate of decrease, which is based on the sampled state of the system. Due to the lack of continuous state monitoring, the system is classified as an open-loop system during the intervals between sampling instances. The algorithm developed and presented here is based on the work found in [10], [26], [27] and [28]. Figure 3.9 presents the main idea behind ST as a block diagram, which highlights that the triggering condition uses the sampled state as an input argument.

The SS representation of the system is revisited:

$$\begin{cases} \dot{x}(t) & = Ax(t) + Bu(t) \\ y(t) & = Cx(t) \end{cases} \tag{3.46}$$

and the sampled version of the linear feedback control law:

$$u(t_k) = Kx(t_k) \tag{3.47}$$

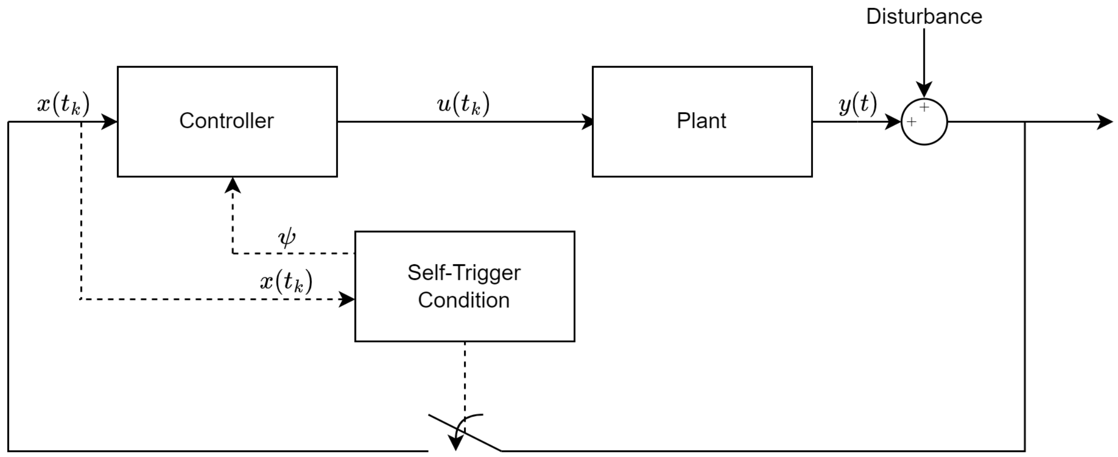


Figure 3.9: Block diagram visually presenting the main idea behind STC.

where the gain matrix, K, is designed in the same way as described in Subsection 3.4.1, by solving the LMI denoted in Equation (3.31). Consider the previously presented quadratic Lyapunov function:

$$V(x(t)) = x^T(t)Px(t) \tag{3.48}$$

where P is a symmetric and positive definite matrix.The main idea for the implementation of STC in this thesis is based on comparing a Lyapunov function of a computed state trajectory with an exponentially decreasing function [28]. A copy of the dynamics of the system, denoted by Equation (3.46), with the same control input, as denoted in Equation (3.47), is used to compute the expected trajectory of the plant [28]:

$$\dot{\xi}(t) = A\xi(t) + BKx(t_k) \tag{3.49}$$

The quadratic Lyapunov function presented in Equation (3.48) of the closed-loop reference system presented in Equation (3.49) is:

$$V(\xi(t)) = \xi^T(t)P\xi(t) \tag{3.50}$$

The exponentially decreasing function is defined as [28]:

$$V_{exp}(t) = V(x(t_k))e^{-\lambda\tau} \tag{3.51}$$

where $V(x(t_k))$ is the value of the sampled quadratic Lyapunov function shown in Equation (3.48) at time t_k, $\tau = t - t_k$ is the time between the current time t and the time of the current update and sampling instance t_k, and λ is the desired decay rate of the system. The decay rate is defined as the smallest value found by considering the absolute value of the real part of the eigenvalues of the system [27]. In other words, it is the eigenvalue of the system furthest to the right in the left half plane. This is also referred to as the dominant eigenvalue of the system [17]. This is formally written as:

$$\lambda = \min |real\{\lambda_i(A + BK)\}| \tag{3.52}$$

where $\lambda(A + BK)$ is the eigenvalue of $A + BK$. The reason for this choice of desired decay rate is that it is desired that the STC implementation has the same decay rate as a traditional periodical feedback system. Which has been designed by one of the Equations (3.20) or (3.31). The triggering condition is based on the comparison between the Lyapunov function of the closed-loop reference system, presented in Equation (3.50), and the exponentially decreasing function, presented in Equation (3.51), which is formalised as:

$$V(\xi(t)) \leq V_{exp}(t) \tag{3.53}$$

The inequality presented in Equation (3.53) is rewritten to be represented by the function $h(t, x(t))$:

$$h(t, x(t)) = V(\xi(t)) - V_{exp}(t) \leq 0 \tag{3.54}$$

To find the next update and sampling instance for the system, the function presented in Equation (3.54) is simulated inside the controller at the time instance of which the control action is updated and a sampling of the state is conducted until inequality is violated. For this to be implemented, the time variable t needs to be changed to the emulated time variable τ, which results in the following [28]:

$$h(\tau, \xi(\tau)) = V(\xi(\tau)) - V_{exp}(\tau) \leq 0 \tag{3.55}$$

As previously stated, the goal of the presented STC strategy is to compute when a new update and sampling instance of the system is required. This can formally be written as[28]:

$$\tau_v = \max\{\tau \leq \tau_{\max} | h(\tau, \xi(\tau)) \leq 0\} \tag{3.56}$$

where the output of the equation is time until the next sampling and update instance, τ_v. τ_v is the max value of τ, below a design parameter, $\tau_{\max}$, that guarantees the maximum time between update and sampling instances, given that the inequality $h(\tau, \xi(\tau))$ of Equation (3.55) has not been violated. The time instant of the next update and sampling instance is given by the sum of the time at the current update and sampling instance, and the computed time until the next update and sampling instance, τ_v:

$$t_{k+1} = t_k + \tau_v \tag{3.57}$$

3.5.1 Implementation in *SIMULINK*

The inequality presented in Equation (3.54) is implemented in *SIMULINK* using the block `MATLAB Function`. The exponential decreasing function, introduced in Equation (3.51), is implemented by sampling the value of the Lyapunov function at time t_k and multiplying it with the exponential term $e^{-\lambda\tau}$. The Lyapunov function computed from the system, which is a copy of the plant dynamics, introduced in Equation (3.50), requires further steps in order to be implemented in the controller. The first step to this implementation is to compute the state trajectory of Equation (3.49) with the time dependency of the emulated time, τ, inside the controller. Previously in Section 3.2, the state trajectory of a system was presented as:

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_0^t e^{A(t-s)}Bu(s)ds \tag{3.58}$$

The notations of the state trajectory, in Equation (3.58), are changed to better suit the current problem, which is the state trajectory of the system, denoted in Equation (3.49):

$$\xi(\tau) = e^{A\tau}x(t_k) + e^{A\tau}\int_0^\tau e^{-As}ds\ Bu(t_k) \tag{3.59}$$

The challenge of implementing Equation (3.59) is the implementation of the integral. The integral has to be numerically approximated, which can be done by applying Tustin approximation as previously described in Subsection 3.2.1. To apply the Tustin approximation, the integral of Equation (3.59) has to be split into two parts:

$$\int_0^\tau e^{-As}ds = \underbrace{\int_0^{\tau-T_s} e^{-As}ds}_{\int(Previous\ Iteration)} + \underbrace{\int_{\tau-T_s}^{\tau} e^{-As}ds}_{\int(Current\ Iteration)} \tag{3.60}$$

where T_s represents the time step duration, which serves as a parameter for quantifying the temporal interval between successive computations associated with estimating the state trajectory. The computational complexity of the STC strategy is linked to the parameter T_s, in the sense that if T_s is chosen to be the same as the minimum update and sampling time, the controller will do a computation to predict the state trajectory as many times as the control action would have been updated in the traditional feedback control strategy. If T_s is specified to be the same as the minimum update and sampling interval, the STC strategy will only save resources for the sensors of the system in terms of bandwidth usage and processing of sampled data. If T_s is specified to be larger than the minimum update and sampling interval, additional resources will be saved in terms of computational power. $\int(Previous\ Iteration)$ is the integral from 0 to the previous emulated time $(\tau - T_s)$ and $\int(Current\ Iteration)$ is the integral from the previous time

$(\tau - T_s)$ to the current emulated time τ. Equation (3.6) is used to approximate the integral which is denoted as $\int(Current\ Iteration)$ as follows:

$$\int_{t_n}^{t_{n+1}} f(t,y(t))dt \approx \frac{h}{2}[f(t_n, y(t_n) + f(t_{n+1}, y(t_{n+1})]$$
$$\Rightarrow \int_{\tau-T_s}^{\tau} e^{-As}ds \approx \frac{T_s}{2}[e^{-A(\tau-T_s)} + e^{-A\tau}] \tag{3.61}$$

By substituting the Tustin approximation of $\int(Current\ Iteration)$ into Equation (3.59) the following expression is found:

$$\xi(\tau) = e^{A\tau}x(t_k) + e^{A\tau}\left[\xi(\tau - T_s) + \frac{T_s}{2}[e^{-A(\tau-T_s)} + e^{-A\tau}]\right] Bu(t_k) \tag{3.62}$$

where $\xi(\tau - T_s)$ is the expression previously considered as $\int(Previous\ Iteration)$. Equation (3.62) is the Tustin approximation of Equation (3.59), which can be implemented in *SIMULINK* using the block `MATLAB Function`. Furthermore, Equation (3.62) is implemented as a factor of the Lyapunov function presented in Equation (3.50), which is a term of the inequality denoted in Equation (3.54). Appendix E.5 contains a Listing with the implementation of Equation (3.62) and Equation (3.57), which are the triggering condition and the formulation used to find the next update and sampling instance of the system.

3.5.2 Example: STC, Triggering Condition

Figure 3.10 provides an example of how the triggering condition, presented in Equation (3.55), works in practice. In the left-hand side subplot, the exponentially decreasing function is represented by the orange lines, and the "lobes" in blue are the computed Lyapunov function of the state trajectory. It is important to note that the first data point of each lobe is based on the true Lyapunov function of the system at the corresponding time instance, while the remainder of the lobe is estimated using Tustin approximation to predict the Lyapunov function of the state trajectory, as described by substituting Equation (3.62) into Equation (3.50). For this reason, the last data point of each "lobe" does not overlap perfectly with the start of the next one due to errors introduced by the Tustin approximation. Normally, a solution to reduce the error is to break down the integrals over smaller steps, i.e. smaller T_s, but this results in more computational resources being spent, which STC aims to reduce in the first place. In the right-hand side subplot, the triggering condition, presented in Equation (3.55), $h(\tau,x(t_k))$, is plotted. From this plot, it can be observed that a new "lobe" starts every time a value greater than zero appears. This is because when a lobe reaches a value greater than zero, the triggering condition, denoted in Equation (3.55), is violated, which means that the next update and sampling instance is found. Which was formally introduced in Equation (3.56) and Equation (3.57).

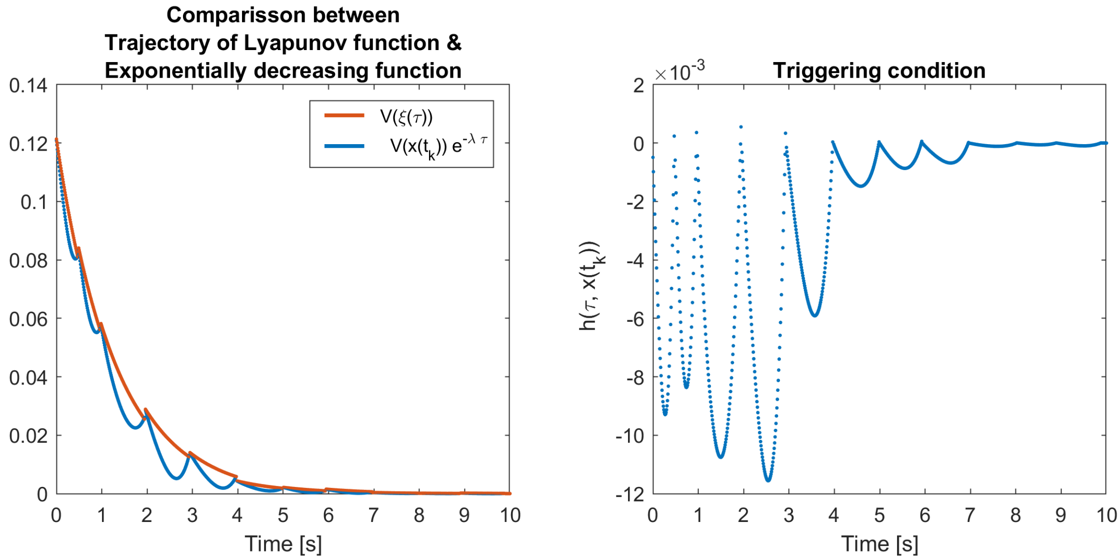


Figure 3.10: Comparison between the computed trajectory of the Lyapunov function and the exponentially decreasing function. Each lobe corresponds to an update and sampling instance. The individual lobe does not end exactly where the next one starts, and this is due to computation errors accumulated because Tustin discretisation is used to estimate the state trajectory of the system.

Chapter 4

Experiments and Results

This chapter is divided into three main sections, where each Section has different setups with different algorithms and parameters:

1. **ETC with the initial gain matrix:** Implementation of the algorithm presented in Section 3.4. Where the gain matrix is computed according to the solution of the LMI defined in Equation (3.20). This means that the system's $\mathbb{D}$-region has not been altered.
2. **ETC with an improved gain matrix:** The algorithm presented in Section 3.4 is used together with a gain matrix computed according to the solution of the LMI presented in Equation (3.31) as introduced in Subsection 3.4.1. Additionally, the algorithm presented in Subsection 3.4.2, which further develops the ETC strategy to allow for reference tracking, is tested.
3. **STC with an improved gain matrix:** Which is the implementation of the STC algorithm presented in Section 3.5 with a gain matrix computed according to the solution of the LMI presented in Equation (3.31) as introduced in Subsection 3.4.1.

The three main sections contain one or more experiments. There are three types of experiments:

1. **Initial conditions:** The system is initialised from a position where the angle of all the axes is different from zero, and the goal is for the system to bring the angles of all the axes to zero degrees. I.e. testing the ability of the system to self-stabilise. The ability to self-stabilise is important because this type of problem often appears from disturbances in drone systems. One such example is a drone affected by a gust of wind. In this case, all the angles of the axes could be driven away from

the desired angle, and by self-stabilising, the drone brings all the axes back to the desired angle.

2. **Disturbances:** A one-second pulse of 5 Volts is introduced to each motor separately to test the control strategy's ability to self-stabilise when exposed to disturbances.

3. **Reference tracking:** In Subsection 3.4.2, it was discussed that driving the different axes to different angles results in propulsion of the system. This is realised by introducing reference tracking, allowing for specified angles to be tracked for each axis.

Table 4.1 provides an overview of what type of experiment has been conducted in each Section.

Table 4.1: Overview of what type of experiments have been conducted in each Section of this Chapter.

	Initial conditions	**Disturbances**	**Ref. tracking**
ETC - Initial K	Experiment 1	Experiment 2	
ETC - Improved K	Experiment 3	Experiment 4	Experiment 5
STC	Experiment 6		

The general structure of the experiments consists of the following results, presented in figures, tables and descriptions:

1. **Periodic control (PC):** Is the result of the traditional PC strategy in simulation, as described in Section 3.1. This is presented first in order to define the *ideal scenario* to compare the results obtained using the ETC and STC strategies in simulation and on the 3 DOF hover.

2. **Simulation of the specified control strategy:** Is the results of the specified control strategy (ETC or STC) in simulation.

3. **Results from the 3 DOF hover the specified control strategy:** Is the results of the specified control strategy (ETC or STC) on the 3 DOF hover.

In each experiment, the same parameters are used across all three cases defined in the list above, which includes the same gain matrix and decay rate parameter. The exception from this structure is Experiment 5 - Reference Tracking, which contains results obtained from the PC strategy on the 3 DOF hover in addition to the structure defined in the list above. This is included in order to compare it to the results from the experiment on the 3 DOF hover because the performance was not as expected. This is discussed to a greater extent in the experiment.

The figures in the experiments contain the following subplots, where subplots 1) $-$ 3) are structured in the same way for all the figures, starting from the uppermost subplot: 1) The output, $y(t)$, of the system. I.e. the angular position of the three axes, yaw pitch and roll. 2) The control action, $u(t)$, of the four motors. I.e. the output voltage of each motor. 3) The Lyapunov function, $V(x(t))$, of the system. This can be thought of as the energy contained in the current state of the system.

After the first three subplots, the figures obtained using the ETC or STC strategies are structured differently from each other. The remaining subplots contain the following:

- **Unique for ETC:** 4) Presents the triggering condition, which is defined as described in Equation (3.28) or Equation (3.45), for ETC and ETC with reference tracking, respectively. 5) Is a binary plot, meaning the values contained in the plot are either logical 1 or 0. If the plot at a time instance is logical 1, this means that the triggering condition at this point of time has been violated, and the controller performs an update to the control action. Otherwise, if the plot at a time instance is logical 0, it means that the triggering condition is not violated, and the control action is kept constant and not updated at this point in time. 6) Displays the interval of time elapsed between control action updates, where the update frequency (UF) also can be observed. Although this data can be viewed in the previous subplot, it is more easily interpreted when plotted against time.
- **Unique for STC:** 4) Is the discretised value of the Lyapunov function, which is used to compute the Lyapunov function of the expected state trajectory and the exponentially decreasing function denoted in Equation (3.55), which is used to compute the next time of which an update and sampling instance is required. 5) Presents the instances in which the control action is updated and sampling of the state is performed. The amplitude of the update instances specifies the time until a new update and sampling instance is required.

4.1 Event-triggered Control - Initial Gain Matrix

The first two experiments, Experiment 1 - Initial Conditions and Experiment 2 - Disturbances, were conducted using a gain matrix K computed by solving the LMI presented in Equation (3.20). The two experiments prove that the performance of the system with the designed gain matrix is not very good in terms of minimising the integral absolute error (IAE) for the three axes of the system. However, the two experiments prove that the ETC strategy works as designed, and the results can be regarded as satisfactory. The solution of the LMI was found using *YALMIP* as the method presented in Subsection 3.3.1 and shown in greater detail in Listing E.1. The computed gain matrix is:

$$K = \begin{bmatrix} 9{,}91 & -1{,}52 & 0 & 6{,}71 & -1{,}03 & 0 \\ 9{,}91 & 1{,}52 & 0 & 6{,}71 & 1{,}03 & 0 \\ -9{,}91 & 0 & -1{,}52 & -6{,}71 & 0 & -1{,}03 \\ -9{,}91 & 0 & 1{,}52 & -6{,}71 & 0 & 1{,}03 \end{bmatrix} \tag{4.1}$$

At an early stage of the project, the decision to use $\sigma = 0{,}997$ was made to achieve the best possible performance in terms of deviation from the desired reference. This value of σ has been kept constant for all the experiments throughout the thesis. In Appendix C, a comparison of three different values of σ is presented in a figure. The figure shows that the performance in terms of the deviation of the state is best in the rightmost column, obtained using $\sigma = 0{,}997$. However, this comes at the cost of a higher UF, which can be observed by considering the leftmost column, obtained using $\sigma = 0{,}333$. In hindsight, it can be discussed if choosing $\sigma = 0{,}997$ for the entire project is the most logical choice since it comes at the cost of a higher UF.

4.1.1 Experiment 1 - Initial Conditions

Quantitative measurements from the experiment are presented in Table 4.2. The first column indicates the type of test, where *Sim.* denotes test in simulation and *Lab.* denotes a test on the 3 DOF hover. The second column corresponds to the figure associated with the test. The columns labelled *yaw*, *pitch*, *roll*, and *total* show the IAE of the specified axis as well as the total of all axes combined. This measurement provides insight into how well the system performs in terms of error. The rightmost column presents the UF of the control action, i.e. how many times the control action is updated per second. The UF is set to 1000 for the traditional PC strategy, which is also the maximum UF that the other two control strategies (ETC and STC) can utilise. It can be discussed whether a UF of 1000 is required for the traditional PC strategy or if comparable results can be attained with a lower UF. However, this UF is chosen to enable the ETC and STC strategies to operate at UF as high as this when required. Each experiment will have a table structured the same way as this one, and the explanation provided here will not be repeated.

Table 4.2: IAE for each axis, the total IAE across all axes, and the UF.

Strategy	Figure	Yaw	Pitch	Roll	Total	UF
PC Sim.	4.1	12,81	16,92	18,37	48,10	1000
ETC Sim.	4.2	11,89	15,91	19,55	47,35	264,30
ETC Lab.	4.3	30,62	19,19	30,51	80,33	159,90

Based on the IAE metric, the ETC strategy in simulations, displayed in Figure 4.2, demonstrates that the performance is on par with the ideal scenario shown in Figure 4.1. However, the ETC provides an advantage because its average UF is lower than the ideal scenario, therefore saving resources in terms of computing power. On the 3 DOF hover, displayed in Figure 4.3, the ETC strategy's performance was worse than the results of simulations when considering the IAE metric. However, it was found that the UF on the 3 DOF hover was lower compared to the simulation. In all of the three figures in this experiment, it can be seen that the Lyapunov function has a decreasing trend, with some oscillations being present in the test conducted on the 3 DOF hover.

This experiment shows that the system is able to successfully self-stabilise from initial conditions different from zero. However, it can be argued that the settling time for all the tests in the experiment is relatively high, in addition to the relatively high UF of the control action, compared to experiments presented later. The reason for this is possibly that the derived gain matrix is weak in terms of its ability to deal with deviations in the feedback loop. Later, in Experiment 3 - Initial Conditions, the same type of experiment as the one conducted here will be repeated with a gain matrix that compensates for deviations to a greater extent.

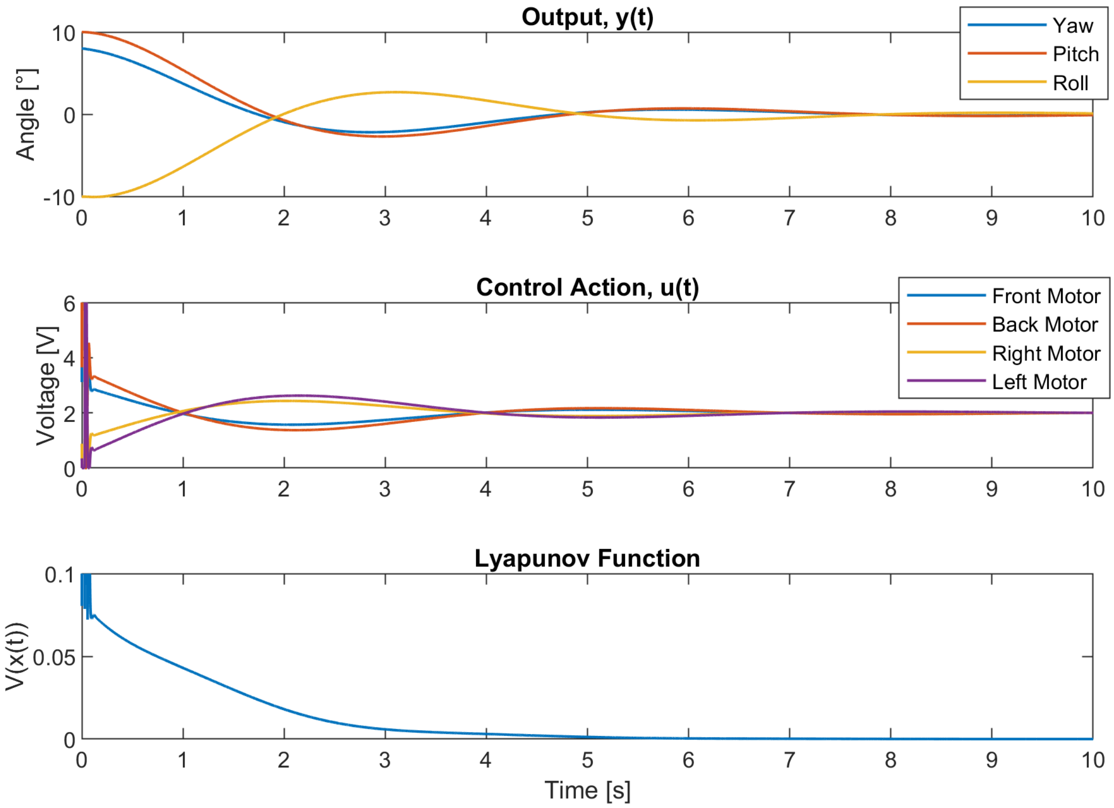


Figure 4.1: In **simulation**, the **self-stabilising ability** of the **PC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.20).

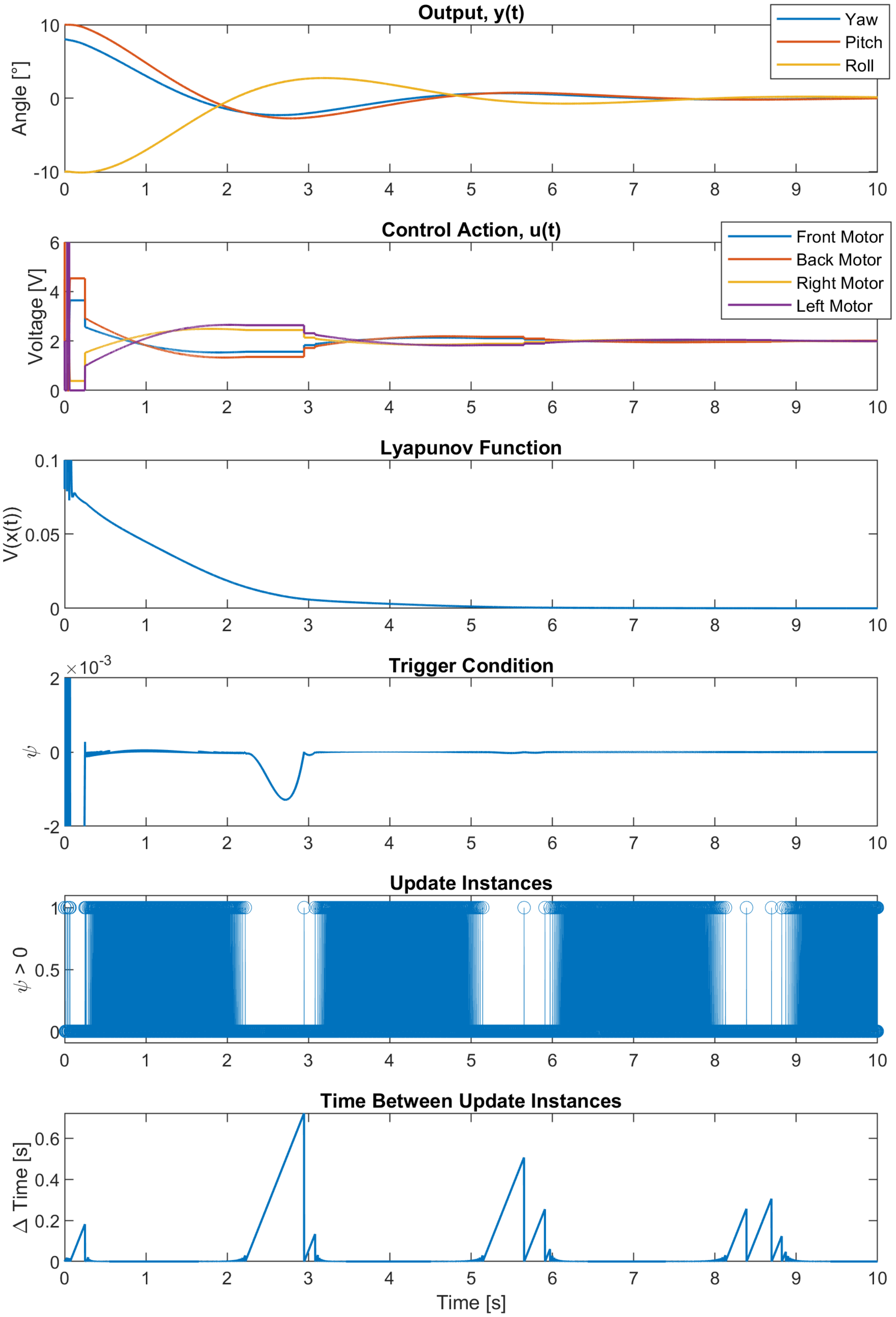


Figure 4.2: In **simulation**, the **self-stabilising ability** of the **ETC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.20).

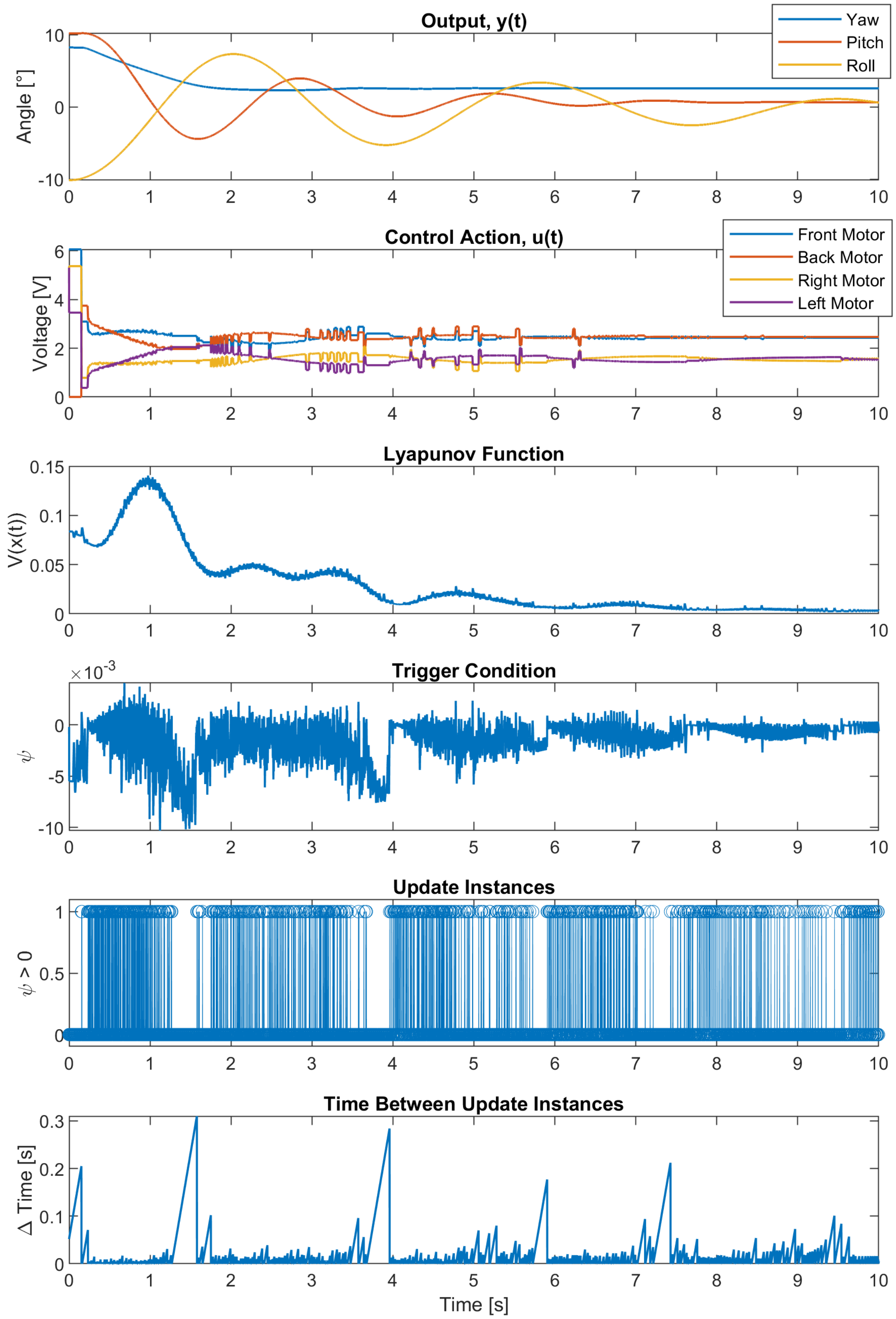


Figure 4.3: On the **3 DOF hover**, the **self-stabilising ability** of the **ETC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.20).

4.1.2 Experiment 2 - Disturbances

When considering the IAE, the performance of the ETC strategy in simulations, as depicted in Figure 4.5, closely resembles that of the conventional PC strategy, depicted in Figure 4.4. This can also be observed in Table 4.3. The UF, both in simulations and on the 3 DOF hover, is similar to the findings in Experiment 1 - Initial Conditions. The IAE performance of the test on the 3 DOF hover, illustrated in Figure 4.6, is better compared to the results obtained from the PC and ETC strategies in simulations. This is likely due to dynamics and disturbances which is not implemented in the simulation model. The decay of the Lyapunov function for all the tests is satisfactory. All the tests show a decaying trend. One issue with the test on the 3 DOF hover is that there are major oscillations present in the system, which, as explained in Experiment 1 - Initial Conditions, likely is due to the gain matrix of the system not being adequate.

Table 4.3: IAE for each axis, the total IAE across all axes, and the UF.

Strategy	Figure	Yaw	Pitch	Roll	Total	UF
PC Sim.	4.4	57,67	375,28	374,79	807,74	1000
ETC Sim.	4.5	62,74	408,18	407,74	878,66	312,57
ETC Lab.	4.6	87,28	49,98	144,84	282,10	133,65

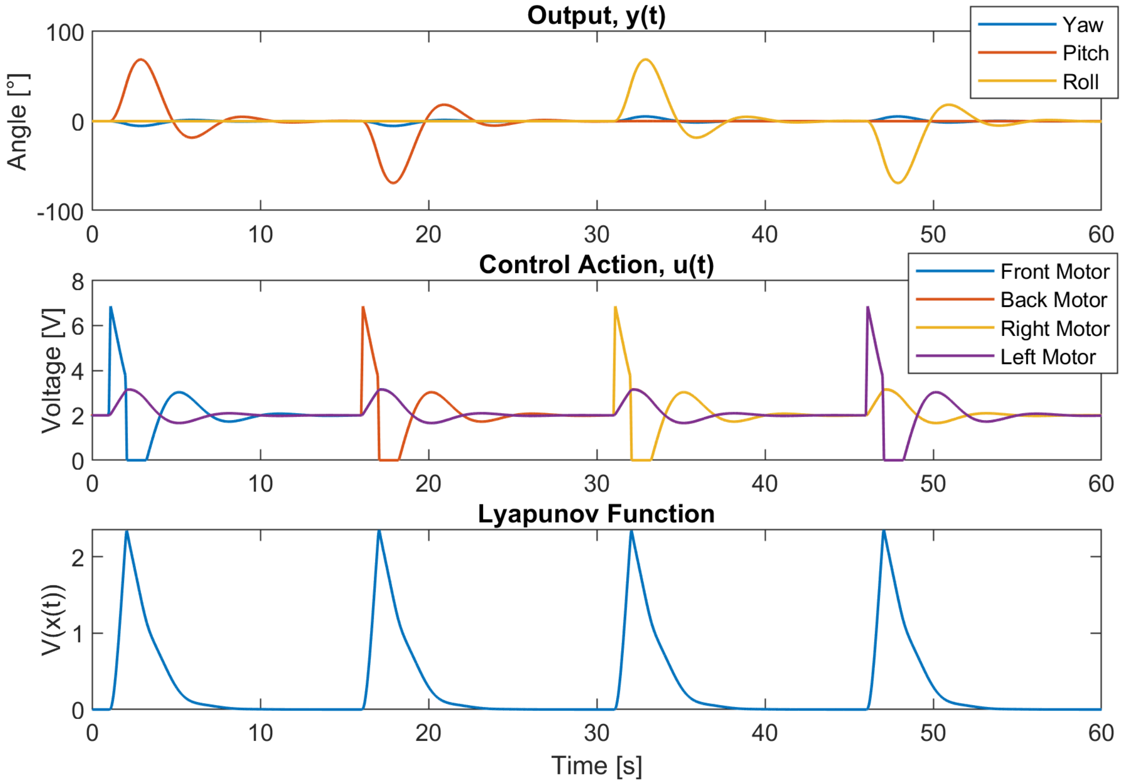


Figure 4.4: In **simulation**, the ability to self-stabilise when exposed to **disturbances** introduced to the motors of the **PC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.20).

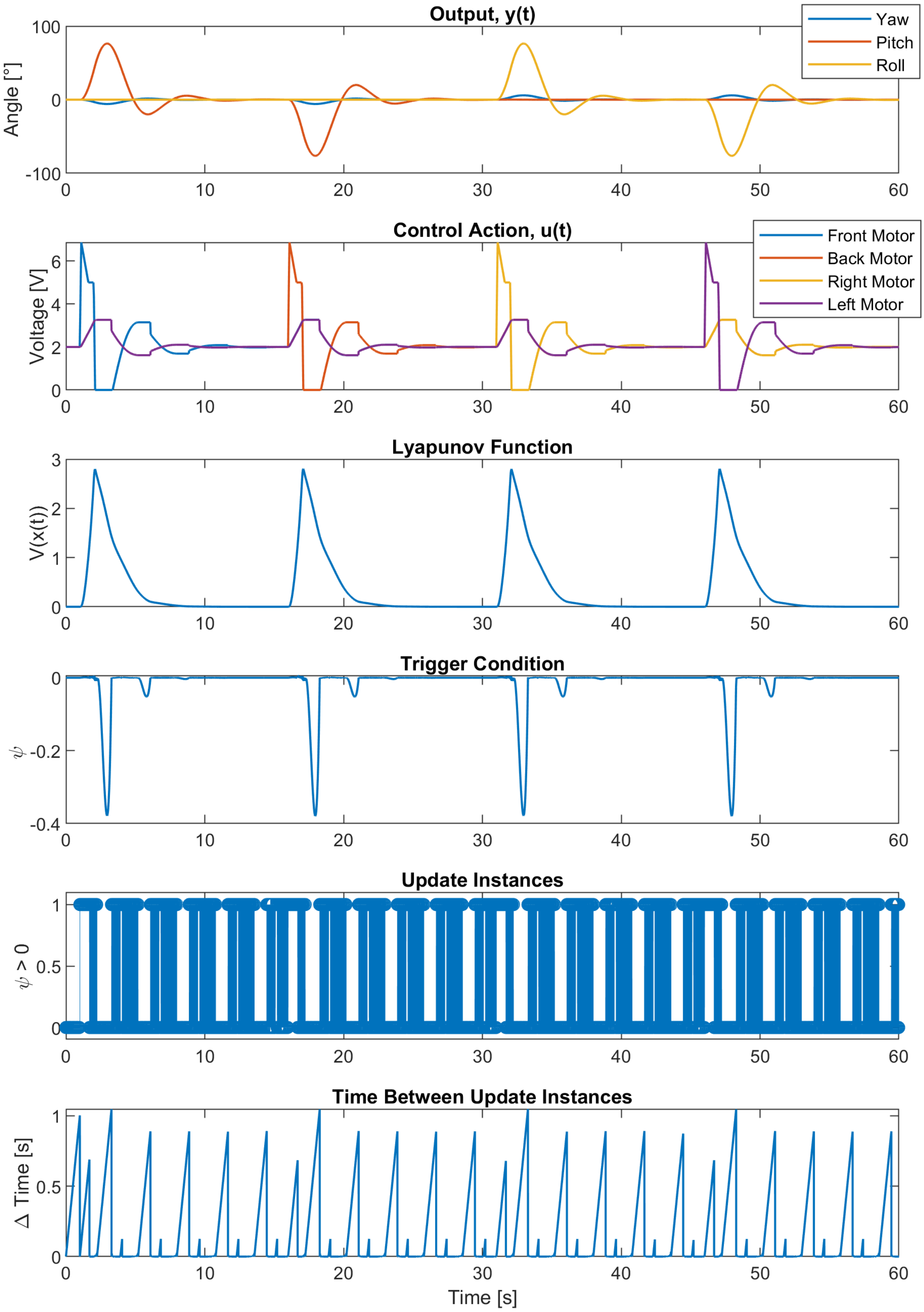


Figure 4.5: In **simulation**, the ability to self-stabilise when exposed to **disturbances** introduced to the motors of the **ETC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.20).

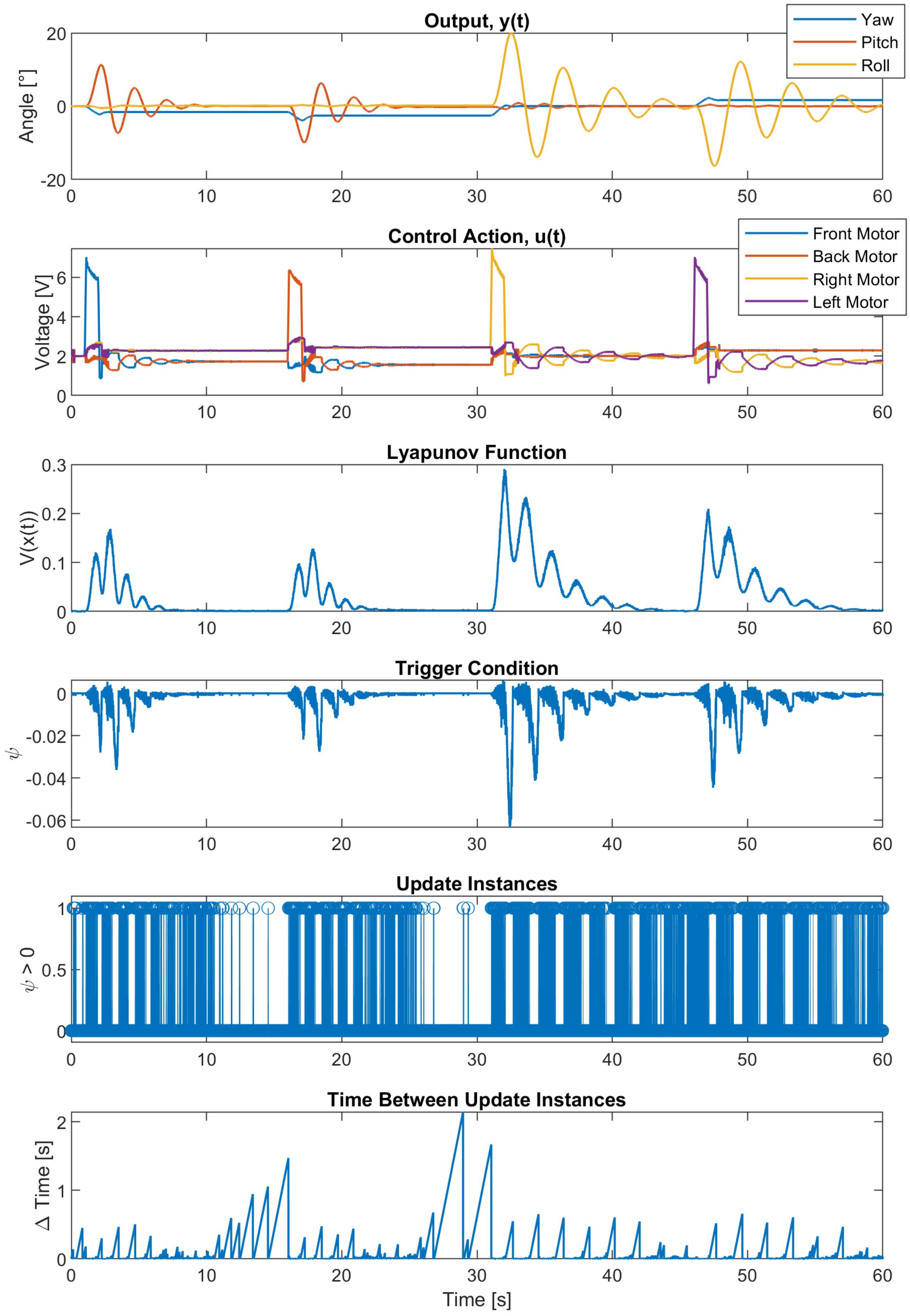


Figure 4.6: On the **3 DOF hover**, the ability to self-stabilise when exposed to **disturbances** introduced to the motors of the **ETC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.20).

4.2 Event-triggered Control - Improved Gain Matrix

The three experiments in this Section have been conducted using an improved gain matrix, K. The gain matrix is computed by solving the LMI presented in Equation (3.31). The solution of the LMI was found using *YALMIP* as shown in Listing E.2. The improved gain matrix is designed by finding the value of α that minimises the IAE of the three axes and the number of times the control action is updated. The best α is found by testing in the type of experiment denoted as *Disturbances*. Figure 4.7 presents the results of varying the value of α. In the left subplot, the total IAE of the three axes is highlighted in blue, alongside the UF of the control action in orange. A weighted sum of the two performance measurements is presented in the right subplot. The weighted sum is computed as:

$$Weighted\ sum = Total\ IAE + 5 \cdot UF \tag{4.2}$$

where the importance of the UF is considered by weighing it such that it is five times more important than the total IAE of the three axes.

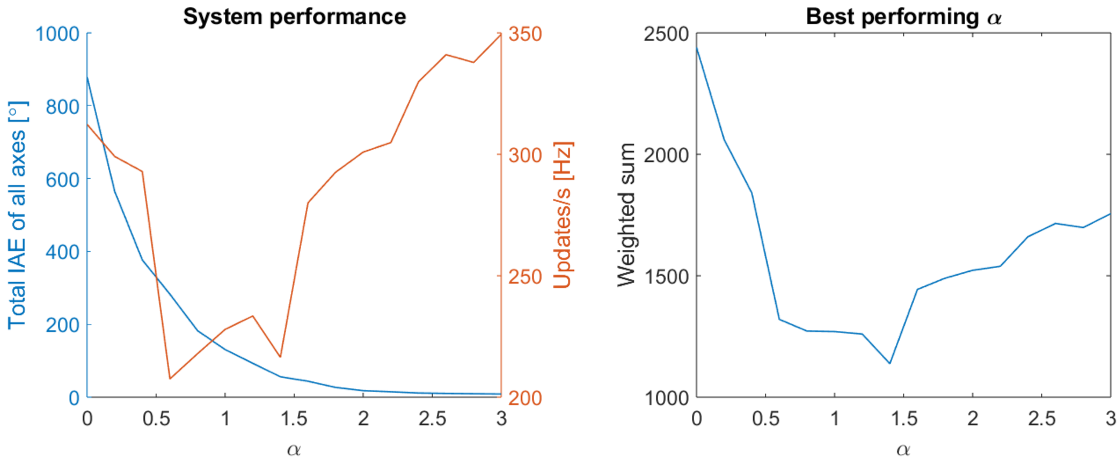


Figure 4.7: Experimentation in simulation to find best suiting α. The best choice, as observed in the right-hand-side subplot, is $\alpha = 1{,}4$.

Solving the LMI denoted in Equation (3.31), with $\alpha = 1{,}4$ results in the gain matrix:

$$K = \begin{bmatrix} 177{,}43 & -27{,}31 & 60{,}29 & -9{,}28 & 0 \\ 177{,}43 & 27{,}31 & 60{,}29 & 9{,}28 & 0 \\ -177{,}43 & 0 & -60{,}29 & 0 & -9{,}28 \\ -177{,}43 & 0 & -60{,}29 & 0 & 9{,}28 \end{bmatrix} \tag{4.3}$$

compared to the gain matrix presented in Equation (4.1), used in the previous two experiments, it can be seen that the elements of the improved gain matrix are greater in absolute value. This implies that for the same deviations, the controller will provide control actions of a greater magnitude.

4.2.1 Experiment 3 - Initial Conditions

The IAE metric of the ETC strategy in simulation, as shown in Figure 4.9, is comparable to the ideal control of the PC strategy, as depicted in Figure 4.8. Furthermore, the IAE metric of the ETC strategy implemented on the 3 DOF hover is significantly better than that of the previous experiment, Experiment 1 - Initial Conditions. The updated gain matrix has also had a positive impact on the UF, with Table 4.4 illustrating a significant decrease compared to Experiment 1 - Initial Conditions. Both in simulations and on the 3 DOF hover, the Lyapunov function is performing well. The improvement in the decay of the Lyapunov function of the ETC strategy on the 3 DOF hover is especially noteworthy as the trend has improved.

Table 4.4: IAE for each axis, the total IAE across all axes, and the UF.

Strategy	**Figure**	**Yaw**	**Pitch**	**Roll**	**Total**	**UF**
PC Sim.	4.8	4,92	4,49	8,57	17,99	1000
ETC Sim.	4.9	4,50	4,87	8,29	17,67	48,00
ETC Lab.	4.10	4,95	13,64	18,34	36,92	183,20

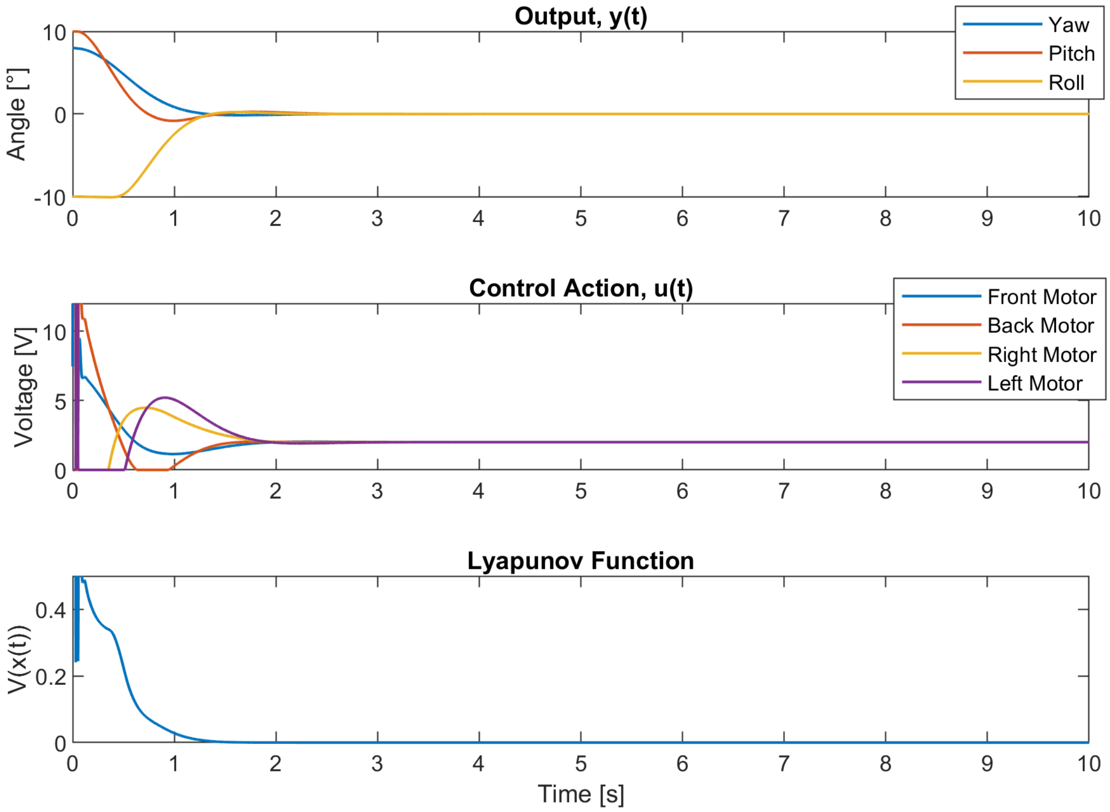


Figure 4.8: In **simulation**, the **self-stabilising ability** of the **PC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 1.4$.

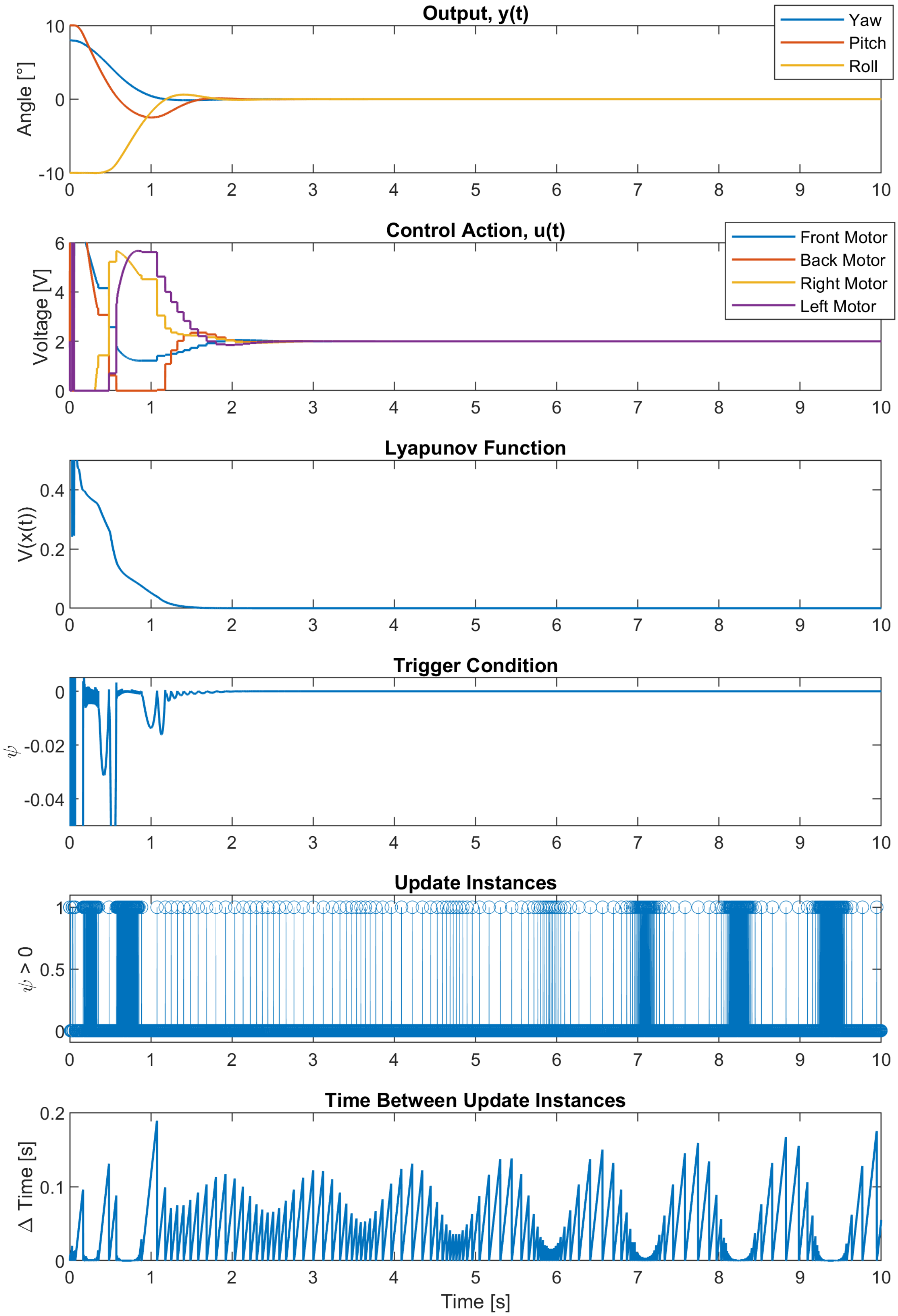


Figure 4.9: In a **simulation environment**, the **self-stabilising ability** of the **ETC strategy** is tested through an experiment using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 1.4$.

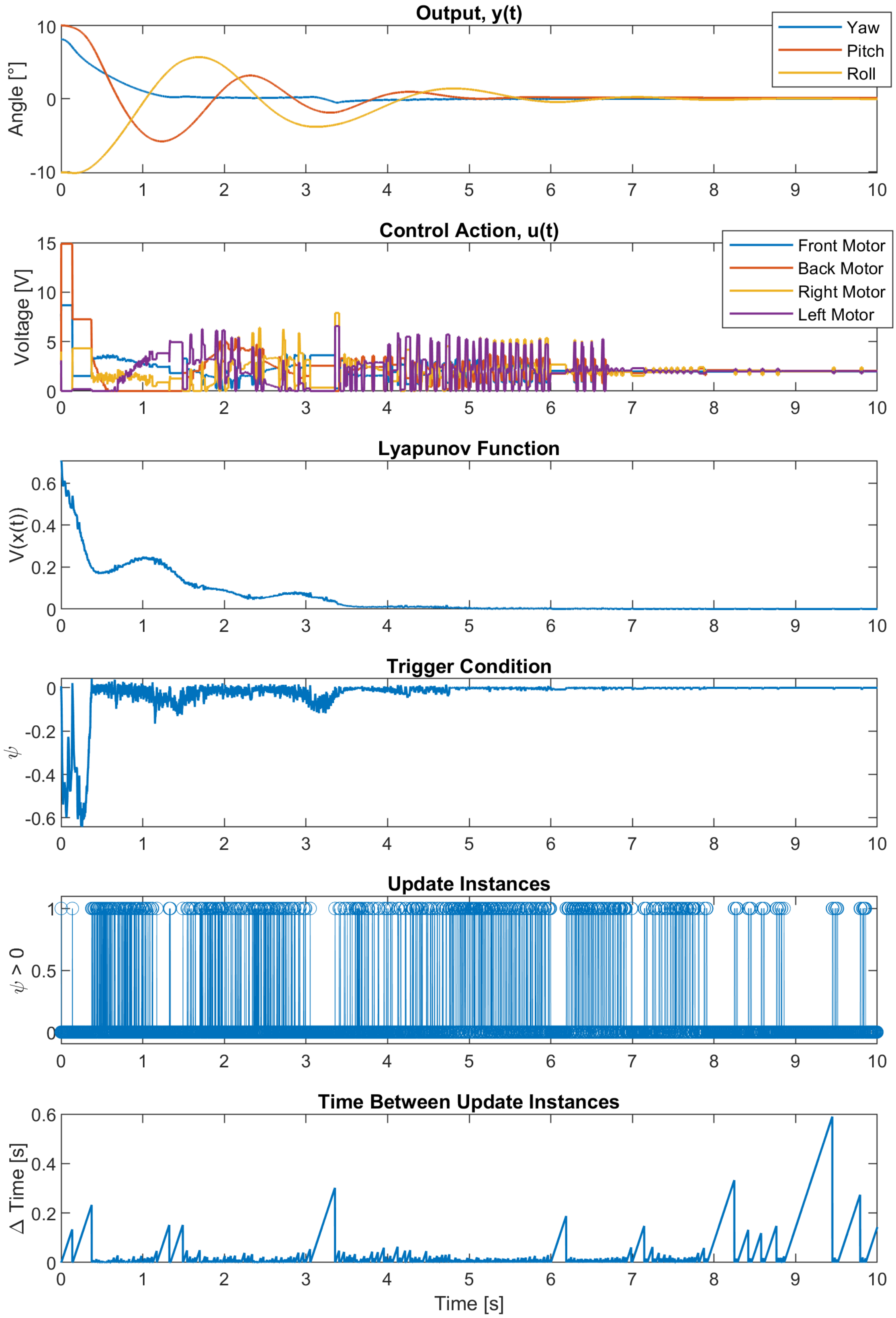


Figure 4.10: On the **3 DOF hover**, the **self-stabilising ability** of the **ETC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 1.4$.

4.2.2 Experiment 4 - Disturbances

The performance of the IAE metric of the ETC strategy in simulations, depicted in Figure 4.12, is comparable to the ideal performance of the PC strategy, depicted in Figure 4.11. Figures 4.12 and 4.13 show that with the improved gain matrix, the ability of the system to self-stabilise when exposed to disturbances is relatively good. From the subplots of the Lyapunov function, it can be observed that the Lyapunov function is swiftly decreasing in all cases. In this experiment, the overall performance of the ETC strategy implemented on the 3 DOF hover observed in Figure 4.13 is especially good. The reason for highlighting this is that it can be observed that the update instances are concentrated to a higher degree where disturbances are introduced, which means that the sampling period is smaller where there are disturbances and higher where there are no disturbances. In Table 4.5, it can be observed that the UF in this experiment is lower compared to Experiment 2 - Disturbances, which conducted the same experiment with the initial gain matrix. Overall it can be concluded that the improved gain matrix works better both in terms of the IAE metric and the UF by lowering both of them.

Table 4.5: IAE for each axis, the total IAE across all axes, and the UF.

Strategy	Figure	Yaw	Pitch	Roll	Total	UF
PC Sim.	4.11	3,50	27,46	27,72	58,68	1000
ETC Sim.	4.12	3,32	26,23	26,38	55,92	197,93
ETC Lab.	4.13	15,93	22,78	45,18	83,88	121,78

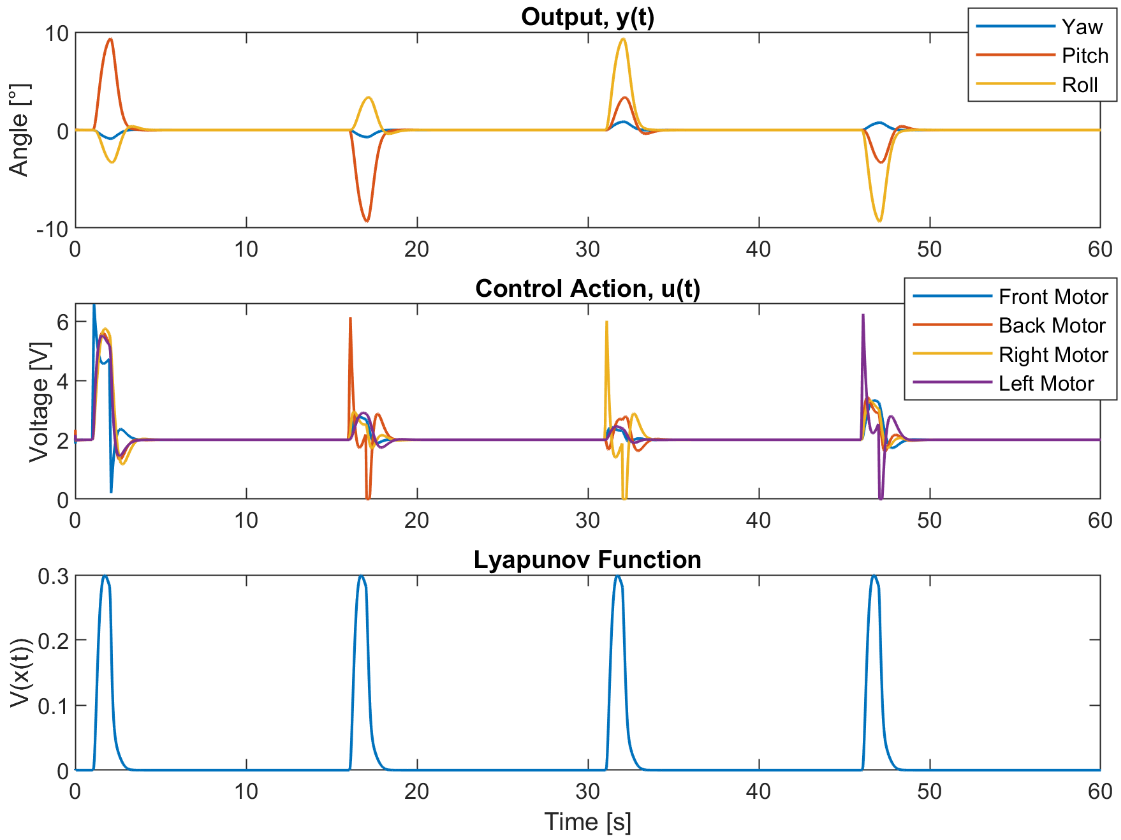


Figure 4.11: In **simulation**, the ability to self-stabilise when exposed to **disturbances** introduced to the motors of the **PC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 1.4$.

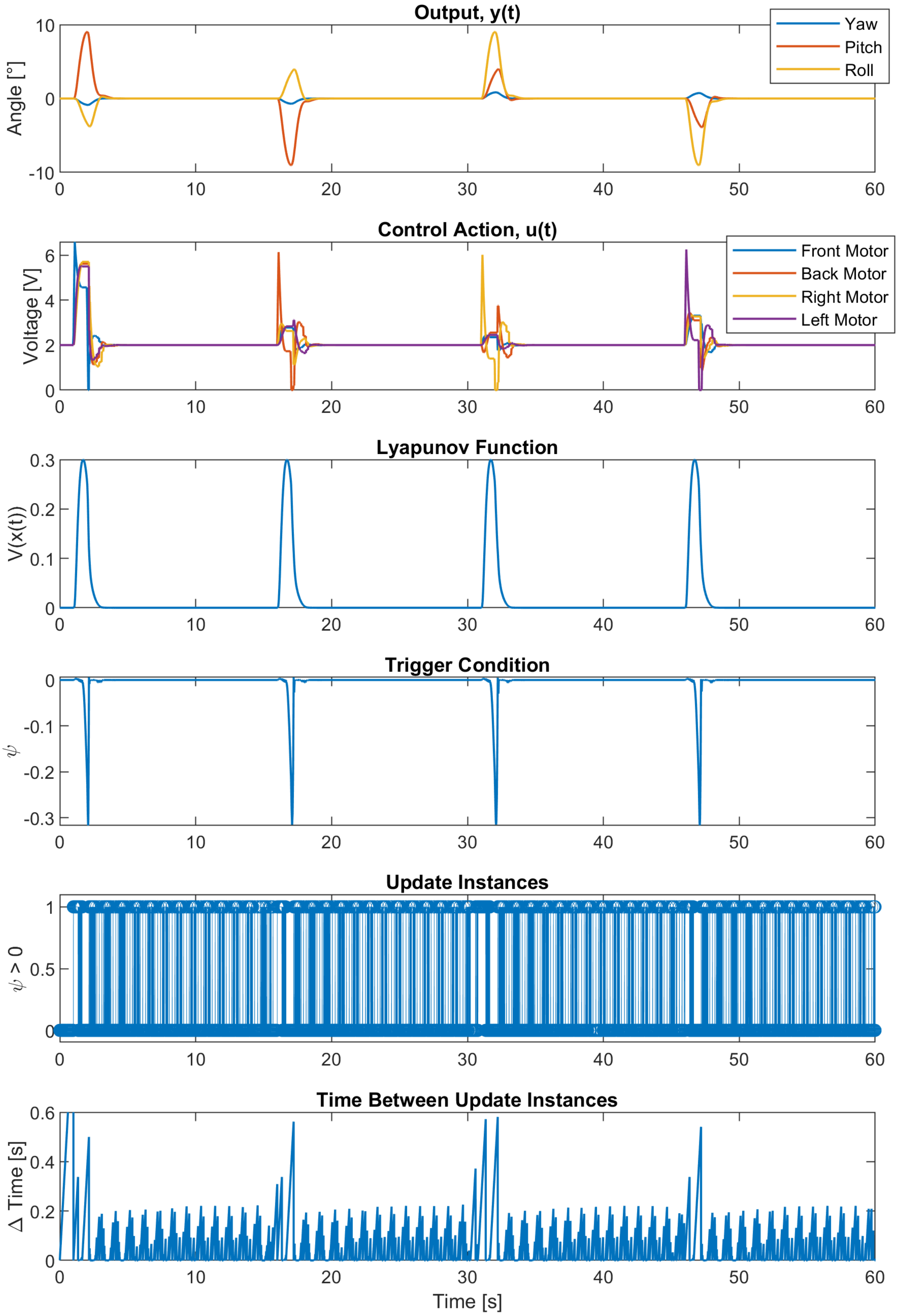


Figure 4.12: In **simulation**, the ability to self-stabilise when exposed to **disturbances** introduced to the motors of the **ETC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 1.4$.

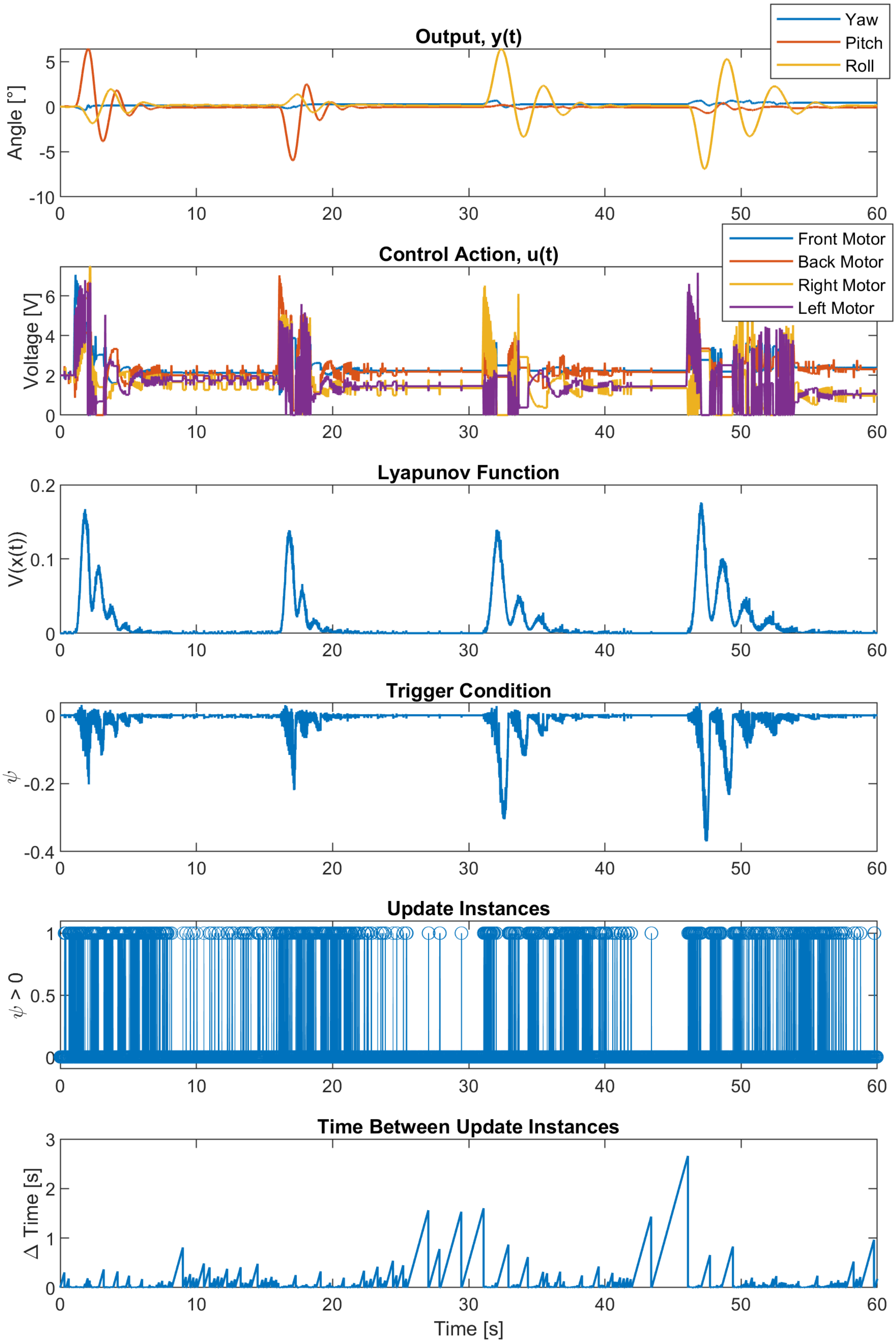


Figure 4.13: On the **3 DOF hover**, the ability to self-stabilise when exposed to **disturbances** introduced to the motors of the **ETC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 1.4$.

4.2.3 Experiment 5 - Reference Tracking

This experiment aims to test the ability of the system to track a reference, achieved by implementing the ETC strategy described in Subsection 3.4.2. Comparisons between the performance of the traditional PC strategy and the ETC strategy in simulations can be reviewed by considering Figures 4.14 and 4.15 or the first two rows of Table 4.6. Considering this, it is observed that the simulation performance is relatively similar between the two strategies, with the exception being the use of computational resources, which is significantly reduced using the ETC strategy.

Table 4.6: IAE for each axis, the total IAE across all axes, and the UF.

Strategy	Figure	Yaw	Pitch	Roll	Total	UF
PC Sim.	4.14	9,35	10,60	10,51	30,46	1000
ETC Sim.	4.15	8,63	10,32	10,24	29,20	168,00
PC Lab.	4.16	34,36	83,00	86,29	203,65	1000
ETC Lab.	4.17	25,88	78,19	73,58	177,65	97,60

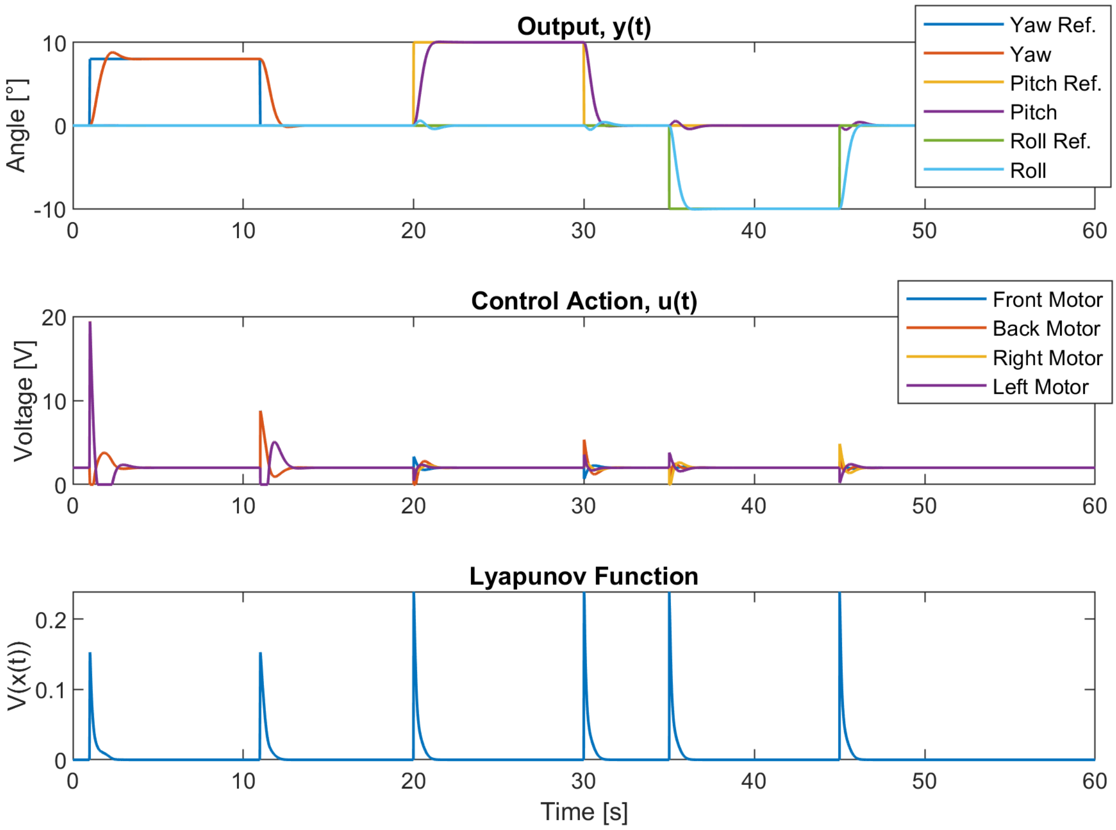


Figure 4.14: In **simulation**, the **reference tracking ability** of the **PC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 1.4$.

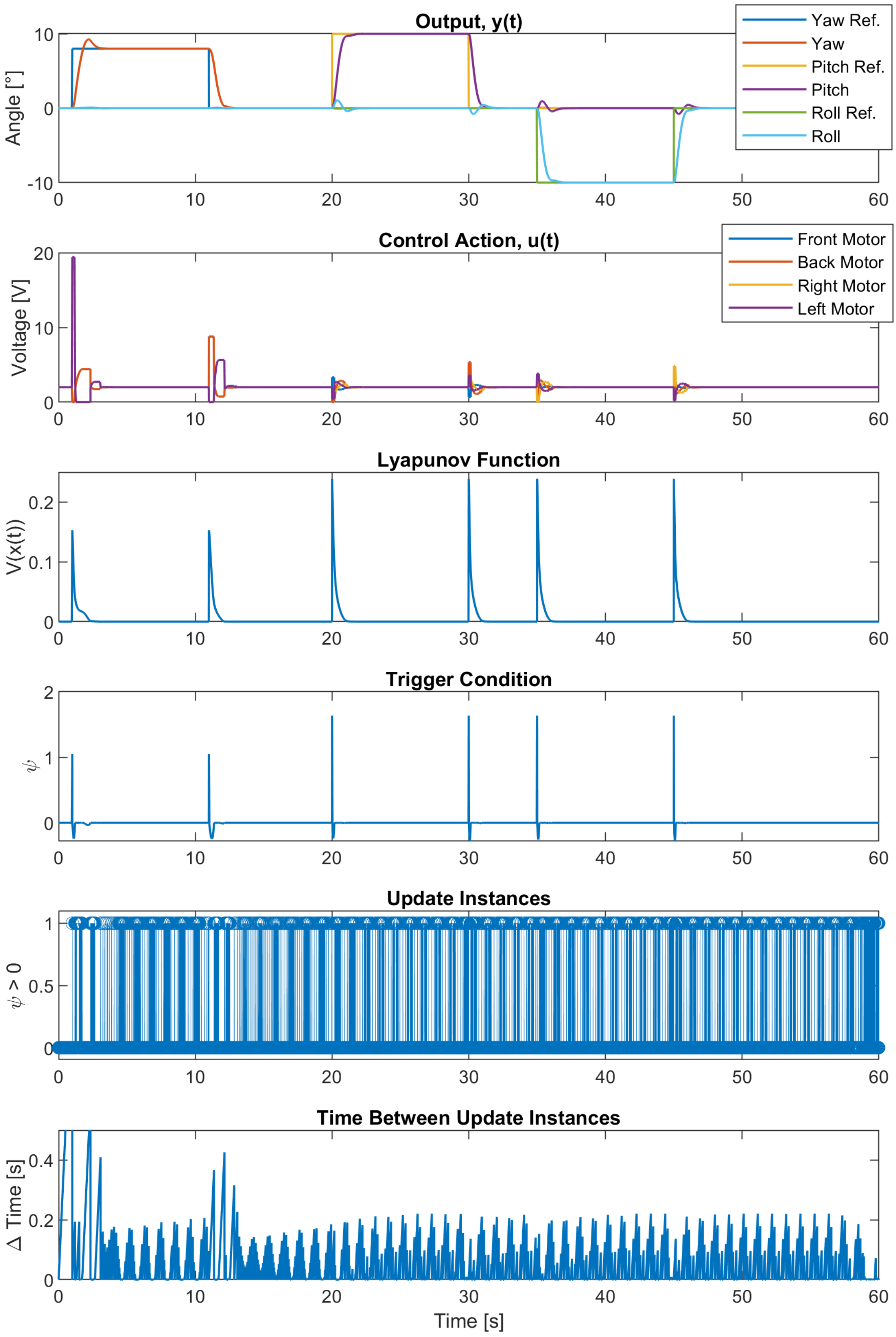


Figure 4.15: In **simulation**, the **reference tracking ability** of the **ETC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 1.4$.

Figure 4.16 and Figure 4.17 depict the tests of the PC strategy and the ETC strategy conducted on the 3 DOF hover. In both cases, the performance is satisfactory when considering the yaw axis, but the performance is poor for the pitch and roll axis, where a significant steady-state error is present. The reason for the poor performance is likely due to the design of the gain matrix. This conclusion can be made because an identical experiment has been conducted for the two strategies, with the exception of how the gain matrix is designed. The gain matrix was designed using the LQR method presented in Section 3.1, where the results are presented in Appendix B. The steady-state error was reduced significantly in the experiment conducted using a gain matrix design by applying the LQR method. However, some steady-state error is still apparent.

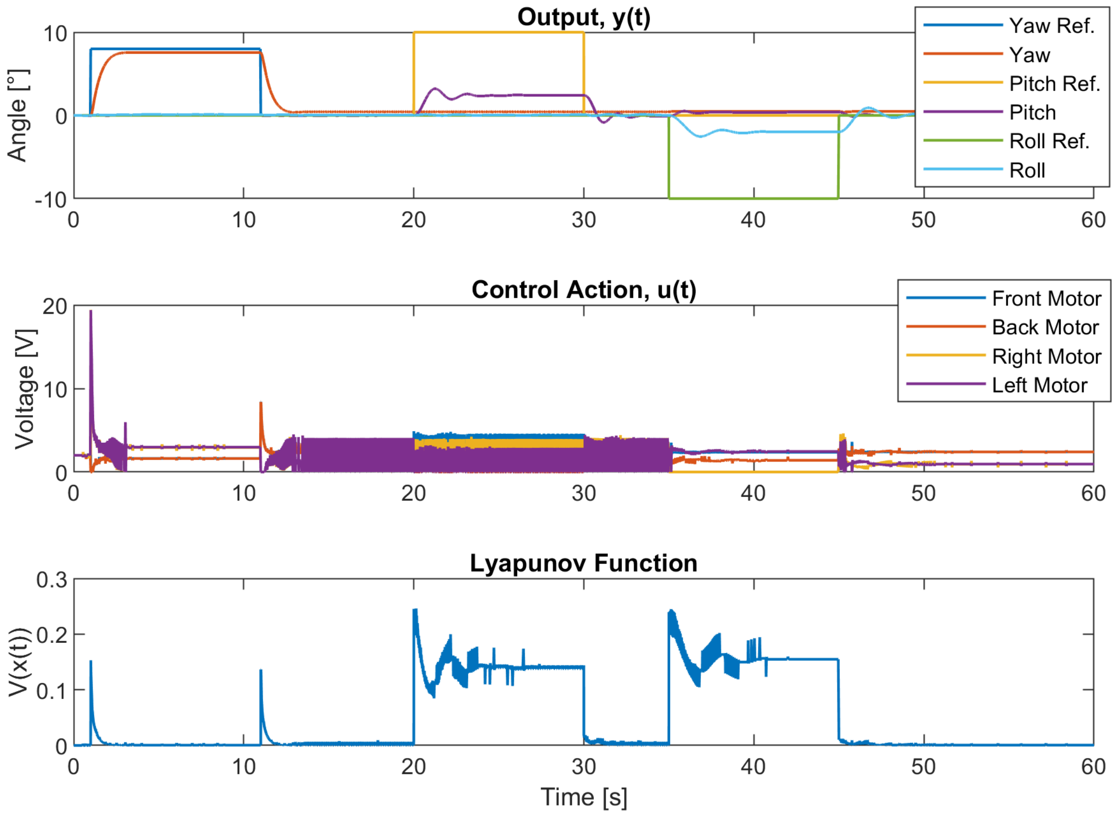


Figure 4.16: On the **3 DOF hover**, the **reference tracking ability** of the **PC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 1.4$.

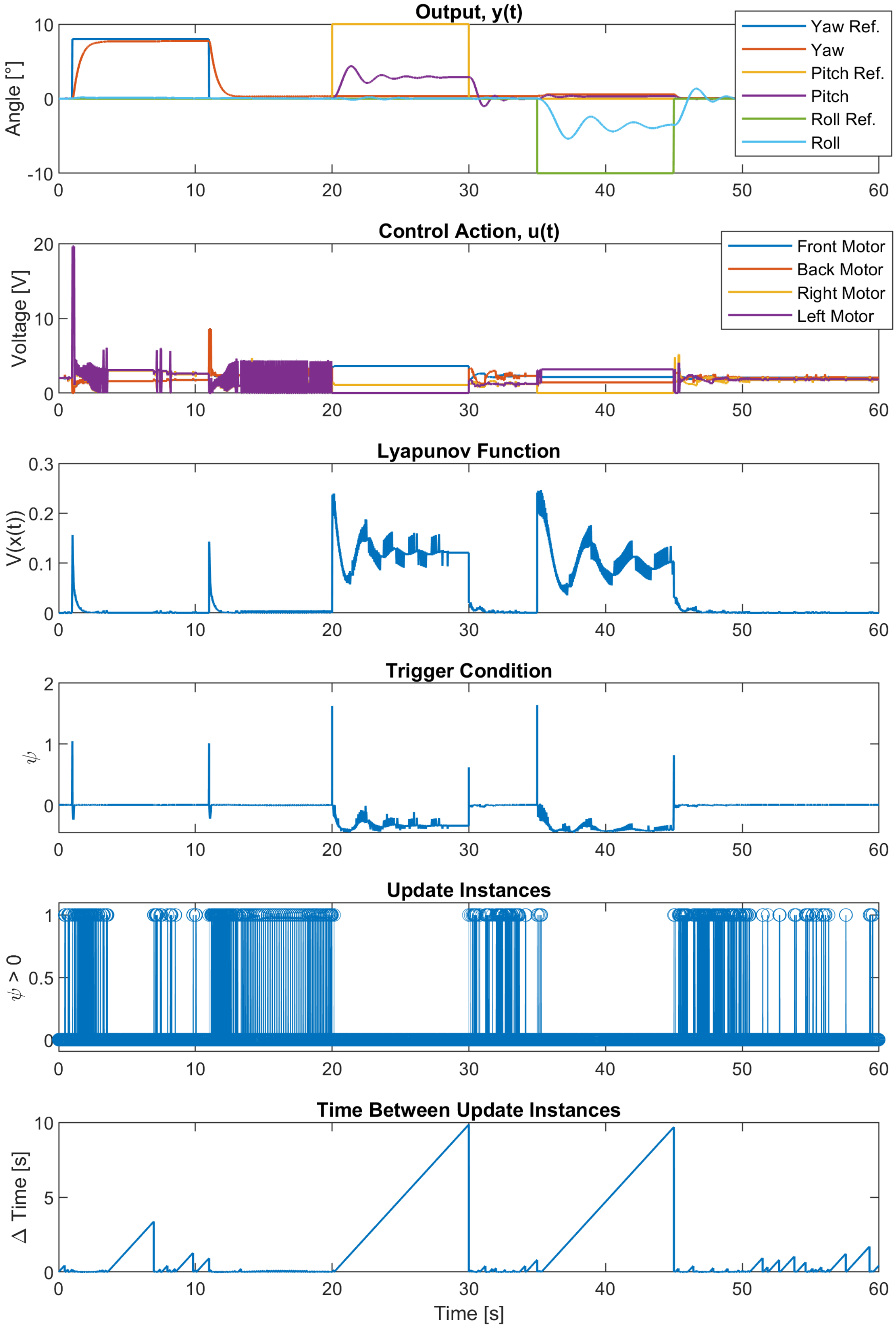


Figure 4.17: On the **3 DOF hover**, the **reference tracking ability** of the **ETC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 1.4$.

4.3 Self-triggered Control

For the STC strategy, the UF denotes the number of times the control action is updated and the number of times the system's state is sampled, which both happen simultaneously. The experiment in this Section has been conducted using an improved gain matrix. The gain matrix has been derived in the same way as described in the previous Section 4.2. To find the best suiting value of α, the experiment denoted as *Initial conditions* was carried out with different values of α, and plotting the resulting IAE, UF and the weighted sum of these two variables, as depicted in Figure 4.18. Because one of the main objectives of the STC algorithm is to reduce the UF of the system, and the magnitude of the IAE metric and UF are very different, the following weighted sum was applied to find the best suiting value of α:

$$Weighted\ sum = Total\ IAE + 20 \cdot UF \tag{4.4}$$

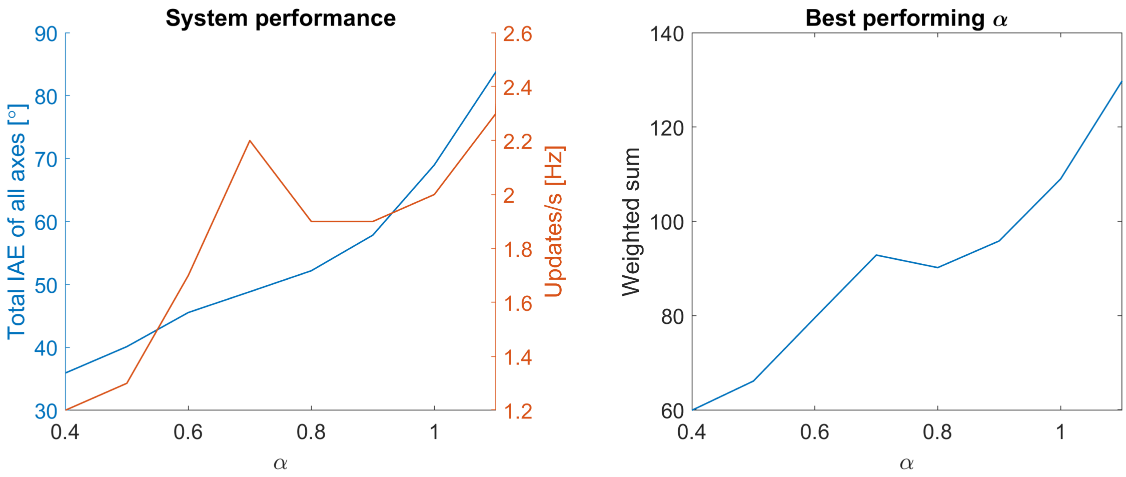


Figure 4.18: Experimentation in simulation to find best suiting α. The best choice, as observed, is $\alpha = 0.4$

The figure only contains values in the range $[0{,}4 \leq \alpha \leq 1{,}1]$ because any values outside this interval resulted in a system that did not diverge to zero within the simulated time of ten seconds. Values smaller than 0,4 failed to reduce the amplitude of the angles, and values greater than 1,1 resulted in oscillations of increasing amplitude.

Solving the LMI denoted in Equation (3.31), with $\alpha = 0{,}4$ results in the gain matrix:

$$K = \begin{bmatrix} 16{,}83 & -2{,}59 & 13{,}17 & -2{,}03 & 0 \\ 16{,}83 & 2{,}59 & 13{,}17 & 2{,}03 & 0 \\ -16{,}83 & 0 & -13{,}17 & 0 & -2{,}03 \\ -16{,}83 & 0 & -13{,}17 & 0 & 2{,}03 \end{bmatrix} \tag{4.5}$$

The time step T_s, which affects the number of computation points of the predicted state trajectory denoted by Equation (3.62), is specified to $T_s = 10ms$ for the STC strategy in this experiment. This results in 100 computations per second when computing the next update and sampling instance.

4.3.1 Experiment 6 - Initial Conditions

In this experiment, the self-stabilising ability of the STC strategy is tested. The figures associated with the first three rows of Table 4.7 are conducted with the gain matrix presented in equation (4.5). The last row of the Table displays the result of a modified version of the STC strategy, depicted in Figure 4.22, which was tuned experimentally to achieve better performance for the STC strategy on the 3 DOF hover.

Table 4.7 shows that the total IAE of the PC and STC strategies in simulations are very similar. The first subplot of Figure 4.19 and Figure 4.20 confirms that the self-stabilising ability of the two methods is similar, bringing all the axes to zero in approximately five seconds in both cases, with the STC strategy being affected by oscillations to a greater extent. This verifies that the STC strategy, at least in simulations, performs satisfactorily. The UF of the STC strategy, both in simulations and on the 3 DOF hover, is very low, which is good in terms of saving computing resources. The defined upper bound, denoted as $\tau_{\max}$ in equation (3.56), was set to ten seconds to check if the system self-stabilised without depending on an upper bound. The results from the simulation depicted in Figure 4.20 proved that, at least in this case, there was no need to define an upper bound. Meanwhile, choosing a lower upper bound, $\tau_{\max}$, could be beneficial for the implementation on the 3 DOF hover. However, this was not defined to keep the same assumptions for the test in simulation and on the 3 DOF hover.

Figure 4.21 presents the self-stabilising ability of the STC strategy tested on the 3 DOF hover. The overall performance depicted in this figure is poor. Despite observing a decaying trend in the Lyapunov function, the system fails to self-stabilise within the ten seconds of the test. The poor performance is likely due to the system being open-loop between update and sampling instances. As a result, the system can not measure disturbance or compensate for unmodelled dynamics affecting it.

Table 4.7: IAE for each axis, the total IAE across all axes, and the UF.

Strategy	Figure	Yaw	Pitch	Roll	Total	UF
PC Sim.	4.19	8,42	11,15	12,79	32,35	1000
STC Sim.	4.20	9,09	11,39	15,44	35,92	1,20
STC Lab.	4.21	13,30	22,72	40,79	76,81	1,80
STC Imp. Lab.	4.22	11,77	11,09	21,05	43,92	7,90

As previously stated, the performance of the STC strategy presented in row three of the Table above and Figure 4.21 was considered to be poor. To achieve a satisfactory performance of the STC strategy on the 3 DOF hover, the parameter α has been tuned experimentally together with changing the computed variable τ_v of Equation (3.56) to $\tau/4$, instead of τ. This is not the best way of compensating for unmodelled dynamics and disturbances. However, for the sake of being able to present results that perform adequately, this was done to compensate for the unmodelled dynamics and disturbances. This resulted in similar performance as in simulation when considering the IAE metric. However, the UF increased by a factor of approximately 6,5.

A better way of handling the issue of the poor performance depicted in Figure 4.21 would likely be to include disturbances in the SS representation presented in Equation (3.46), when designing the STC strategy. The article [27] presents an STC strategy that accounts for disturbances in this fashion.

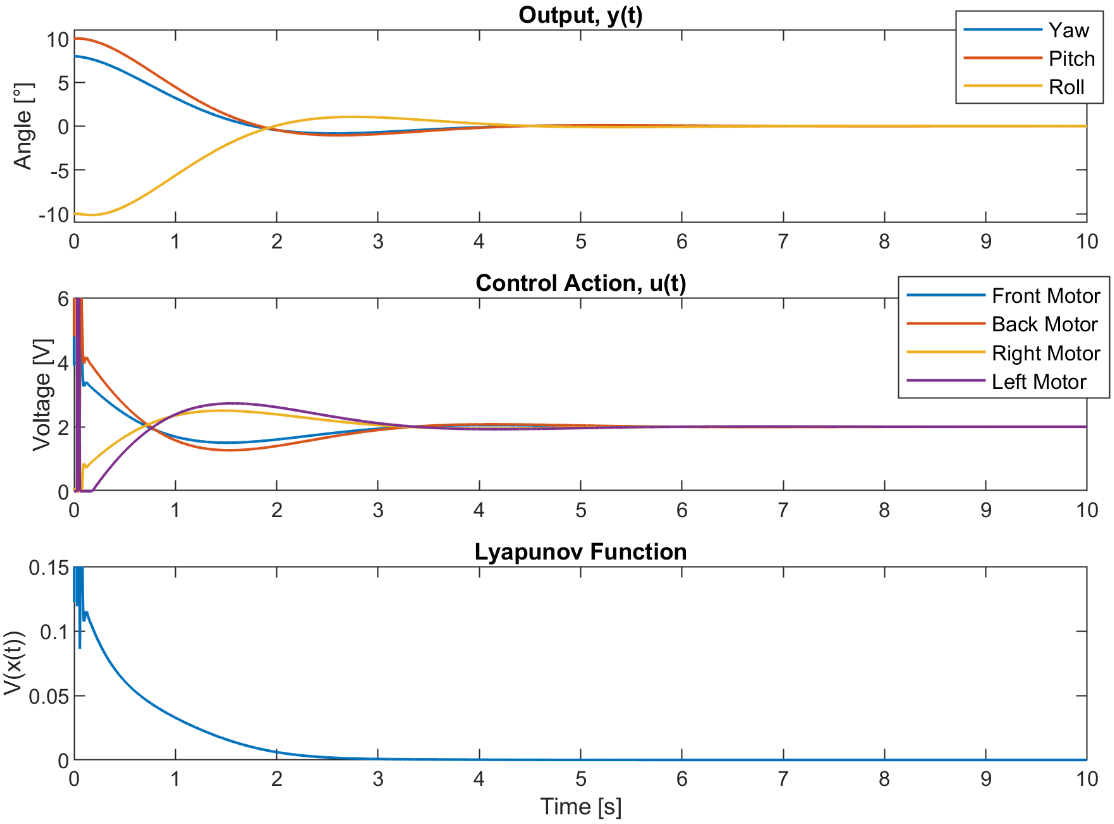


Figure 4.19: In **simulation**, the **self-stabilising ability** of the **PC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 0.4$.

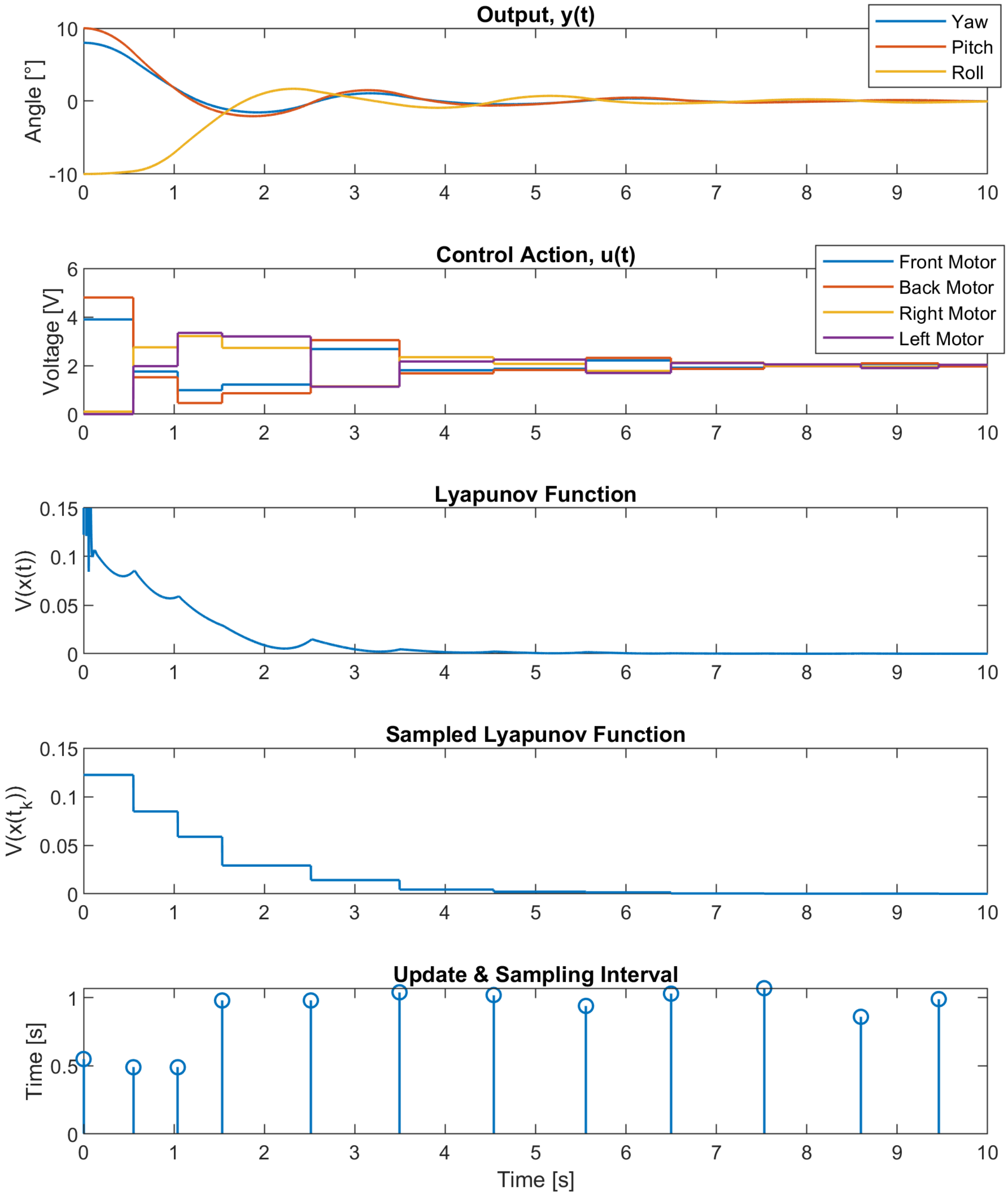


Figure 4.20: In **simulation**, the **self-stabilising ability** of the **STC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 0.4$.

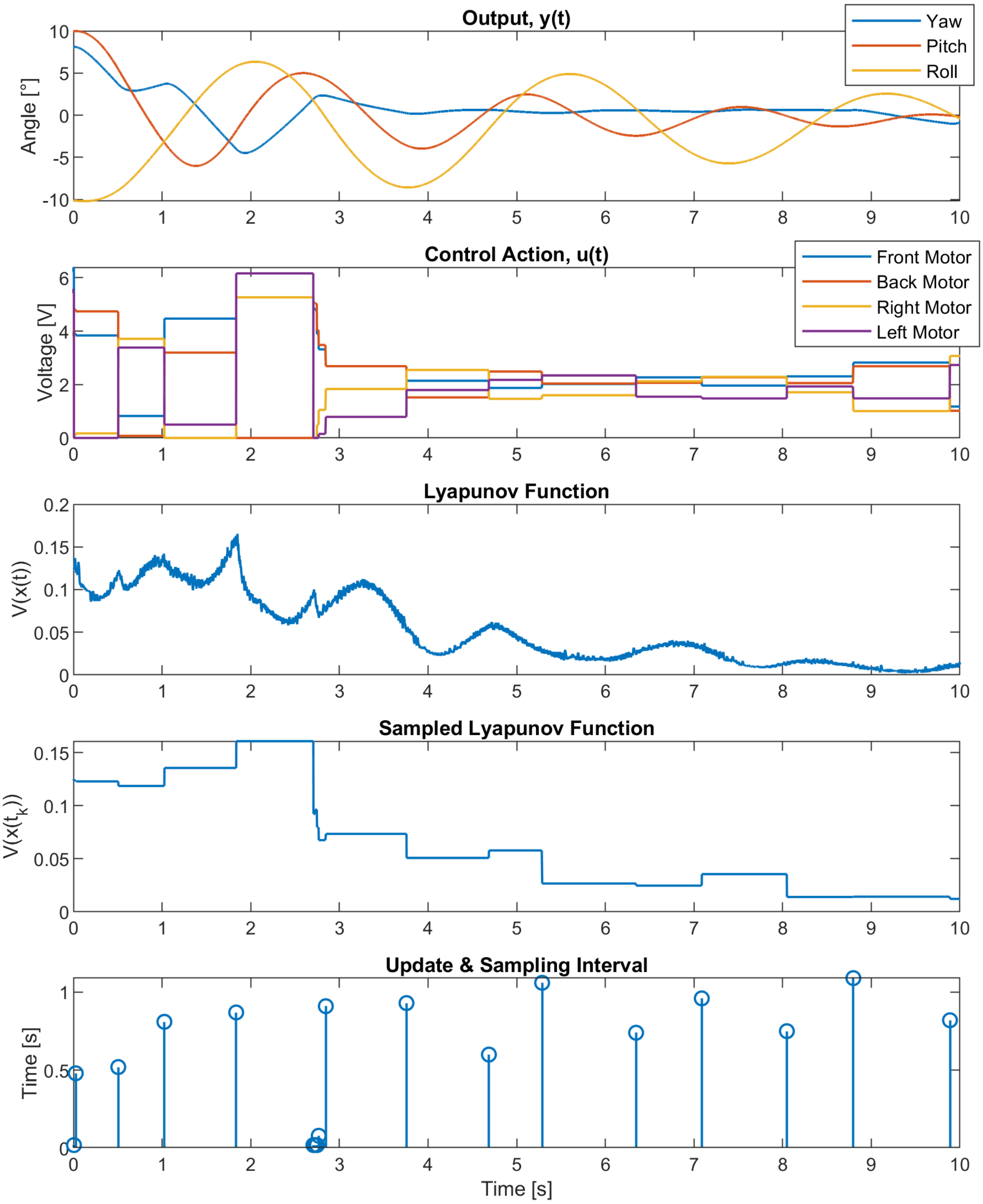


Figure 4.21: On the **3 DOF hover**, the **self-stabilising ability** of the **STC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 0.4$.

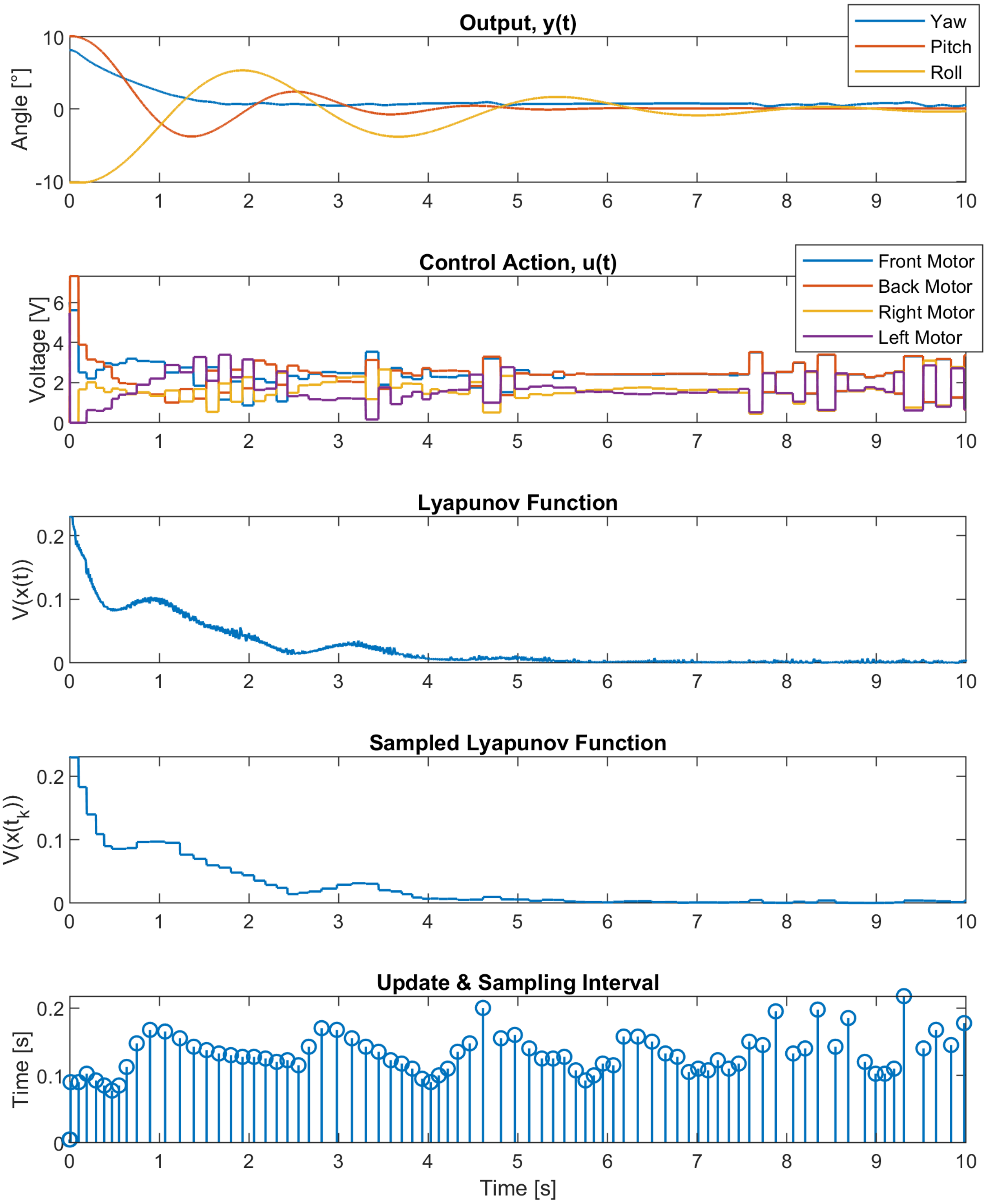


Figure 4.22: On the **3 DOF hover**, the **self-stabilising ability** of the **STC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 0.8$. Additionally, Equation (3.56) has been modified such that the computed next update and sampling instance is $\tau_v/4$

Chapter 5

Discussion

The project description, presented in Section 1.2, states the following:

The goal of the project is to test these two (read ETC and STC) *control strategies on the 3 DOF Hover produced by Quanser which is available in the laboratory at UiS.*

The statement is specified to a greater extent by the outlined specific *activities and objectives*, presented in a bulletin list in the project description in Section 1.2: conducting literature studies, implementing the two control strategies in simulation, and performing experimental validation on the actual 3 DOF hover.

Chapter 3 presents the findings from my literature studies, which have been designed for the 3 DOF hover. Chapter 4 consists of experiments and corresponding results, both in simulation and by experimental validation on the 3 DOF hover. From this, it can be concluded that the objectives of the given project description have been met. In addition to the given objectives of the project description, reference tracking for the ETC strategy was presented and tested in simulation and by experimental validation on the 3 DOF hover. A method of designing the controller/ gain matrix of the two strategies was also presented and tested throughout the experiments of thesis.

5.1 Design of Gain Matrix

Throughout the project, the gain matrix used in the experiments has been designed by solving the LMI denoted by Equation (3.20) or the LMI with the added constraint denoted by Equation (3.31).

The gain matrix utilised in Experiment 1 - Initial Conditions and Experiment 2 - Disturbances was designed by solving the LMI denoted by Equation (3.20). Although successful in proving the effectiveness of the ETC strategy, this proved to produce poor performance in terms of the self-stabilising ability of the system. As a result, the same experiments were conducted with an improved gain matrix designed by solving the LMI denoted by Equation (3.31), containing an additional constraint, as presented in Experiment 3 - Initial Conditions and Experiment 4 - Disturbances. These two experiments proved the improved gain matrix's effectiveness for the system's self-stabilising ability.

The improved gain matrix was also used in Experiment 5 - Reference Tracking, where it produced good results in simulations. Meanwhile, on the 3 DOF hover, a significant steady-state error was present in the pitch and roll axes. Another approach for designing the gain matrix was tested in order to compare it to the existing LMI-based method. This approach was the LQR method, which proved to reduce the steady-state error considerably, although some steady-state error was still present. This raises the question of whether designing the gain matrix based on solving an LMI is the best-suited method for achieving desired performance or if alternative design approaches should be explored to a greater extent.

In Experiment 6 - Initial Conditions, the STC strategy was tested. The gain matrix was found again by solving the LMI denoted in Equation (3.31). Here the system proved to self-stabilise successfully, with acceptable performance in simulations. However, the performance on the 3 DOF hover was poor. The lack of performance here is likely due to unmodelled dynamics and disturbances that have not been accounted for in the design process of the STC strategy and not due to the gain matrix performing inadequately. A solution to this was presented, but it was argued whether this was a good method of dealing with the issue. Another method, more likely to be successful, was discussed. This method includes a term to account for disturbances in the SS model, as presented in the article [27].

5.2 Comparison Between the Control Strategies

All the previous results, presented in Chapter 4, were conducted using different gain matrices in the experiments within each section. To make a fair comparison of the ETC and STC strategies, both should utilise the same gain matrix. The tests presented in Tables 5.1 and 5.2 have been conducted using the same gain matrix, derived from Equation (3.31), with $\alpha = 0{,}4$. The value of α was chosen as it produced acceptable results for the STC strategy in simulation, as well as the ETC strategy has proven to perform adequately for small values of α in simulation, as previously depicted in Figure 4.7. In contrast, higher values of α, which worked well for the ETC strategy, yielded poor results for the STC strategy, as previously depicted in Figure 4.18.

By considering the IAE metric in Tables 5.1 and 5.2, it appears that the performance between the classic PC, ETC and STC strategies are on par with each other, both in simulation and on the 3 DOF hover. Moreover, the UF is the lowest for the STC strategy. This suggests that the STC strategy has the overall best performance, both in simulations and on the 3 DOF hover.

It has been shown previously that the performance of the STC strategy indeed is good in simulation, but that it lacked performance when tested on the 3 DOF hover due to unmodelled dynamics and disturbances. This suggests that the results presented in Table 5.2 are slightly suspicious because the ETC strategy has previously proven to perform well on the 3 DOF hover. By considering the corresponding figures noted in the Table, it can be observed that there are substantial oscillations present for all the strategies on the 3 DOF hover. From this, it is noted that all the strategies perform equally poorly when tested using a gain matrix designed according to Equation (3.31), using $\alpha = 0{,}4$. This suggests that the test on the 3 DOF hover is not a good test to compare the strategies.

Table 5.1: IAE for each axis, the total IAE across all axes, and the UF in simulation.

Simulation

Strategy	Figure	Yaw	Pitch	Roll	Total	UF
PC	4.19	8,42	11,15	12,79	32,35	1000
ETC	D.1	7,20	9,03	15,41	31,64	28,10
STC	4.20	9,09	11,39	15,44	35,92	1,20

Table 5.2: IAE for each axis, the total IAE across all axes, and the UF on the 3 DOF hover.

Laboratory

Strategy	Figure	Yaw	Pitch	Roll	Total	UF
PC	D.2	24,76	11,61	26,89	63,26	1000
ETC	D.3	24,52	14,52	26,85	65,89	159,50
STC	4.21	13,30	22,72	40,78	76,81	1,80

The conclusion drawn from studying the tests presented in Tables 5.1 and 5.2 is that the STC strategy is superior to the ETC strategy in environments where the most pronounced dynamics are modelled and that disturbances are known and can be modelled or that there are no disturbances of significance. While in environments where not all the dynamics are easily modelled, and there are unknown or unmodelled disturbances, the ETC strategy is superior to the STC strategy as long as a suitable gain matrix is utilised. In short, this means that the two strategies are best suited for different types of systems, and one should be chosen over the other for this reason.

5.3 Future Directions

- Explore alternative methods of designing the controller and gain matrix. For instance, investigating the possibility of implementing a non-linear controller, such as a sliding mode controller, see [13] and [29].
- Design of an STC strategy that accounts for disturbances, as presented in article [27] or other methods that account for unmodelled dynamics and disturbances.
- Explore ETC based on output feedback, as presented in paper [30]. This could be interesting because, for many systems, full-state measurements are not available for feedback. Thus it could be interesting to compare the performance of ETC based on output feedback to ETC based on state feedback as presented in this thesis.
- As presented in the previous bulletin, but for STC. An approach for implementing STC based on output feedback is by the introduction of observers, as presented in the paper [28].
- Study and test alternative methods of realising STC and make a comparison to the method in this thesis. The paper [10] presents two methods of implementing STC. The first is the method presented in this thesis, and the second method is called *minimum attention implementation*.

Chapter 6

Conclusion

The overall defined goals of the thesis were to study literature, implementation in simulation and experimental validation of the ETC and STC strategies. All of the defined objectives of the thesis have successfully been completed and presented, and discussed throughout the thesis.

The ETC strategy presented in the theoretical background proved to perform as designed. This was shown in the first two experiments, which required the system to self-stabilise from a non-zero initial position and disturbances, respectively. The two experiments proved that the proposed ETC strategy worked as intended, even though the system's performance only can be classified as adequate due to the long settling times. The performance of the first two experiments gave reason to further develop the design approach of the gain matrix. The first two experiments were conducted again, as presented in experiments three and four, with an improved gain matrix, which performed well. This leads to the conclusion that the ETC strategy designed based on the solution of an LMI defined from a Lyapunov function is a well-suited strategy to self-stabilise the Quanser 3 DOF hover when exposed to non-zero initial positions or disturbances.

An extended version of the first presented ETC strategy was introduced to enable the system to follow a reference signal. This performed well in simulation, but a significant steady-state error was present when implemented on the Quanser 3 DOF hover. Other design parameters were tested for designing the gain matrix without success. As a result, an attempt to test a gain matrix designed using the LQR method was conducted, which proved more successful. This leads to the conclusion that the ETC strategy can be used to realise reference tracking for the Quanser 3 DOF hover successfully. However, the proposed gain matrix designed by solving an LMI defined from a Lyapunov function is unsuitable for this application.

The STC strategy presented in the theoretical background performed as designed when tested in simulation. However, when tested on the Quanser 3 DOF hover, the performance proved to be inadequate. It was concluded that the reason for this is likely caused by the fact that the STC does not actively monitor the system's state, causing the estimated state trajectory of the controller to align poorly against the actual state trajectory of the system due to unmodelled dynamics and disturbances. A solution to this problem was presented, which proved to perform well in compensating for the state error but came at the cost of more updates of the control action and sampling of the system's state. It was concluded that other methods that account for unmodelled dynamics and disturbances should be considered for implementation on the Quanser 3 DOF hover.

Appendix A

Thesis Poster

Event-triggered and self-triggered control of a 3 DOF hover system

Vetle Normann

Department of Electrical Engineering and Computer Science

Universitetet i Stavanger

Background

Traditional control, systems typically depend upon periodic updating of the controller which is designed to be small enough to be disregarded in the design process. One problem with this approach is that many modern control systems are implemented on microcontrollers, which in many cases are restricted by limited resources in terms of computing power and battery energy.

A way of spending less resources is to update the output of the controller more infrequently. There are different ways this can be done. The simplest method is increasing the time between periodic updates, which may significantly affect performance. The most sensible way of approaching this problem is by introducing aperiodic updating of the control action.

In this thesis, two methods of implementing aperiodic updating of the controller are researched, implemented and tested. The two methods are event-triggered control (ETC) and self-triggered control (STC). The implemented algorithm for both methods are based on Lyapunov functions, which guarantees the performance of the controllers.

System

The two control strategies are implemented on the Quanser 3 degree of freedom (DOF) hover, both in a simulation environment and on the physical equipment available in the laboratory of UiS. The 3 DOF hover is a stationary quadcopter with 3 DOF, yaw, pitch, and roll, which is configured to be controlled by Matlab and/ or Simulink. The 3 DOF hover consists of four propellers driven by individual DC motors, mounted on a planar frame which is coupled to the stationary base pedestal. The angles of the three axes are controlled by the propellers and measured by high-resolution encoders. The transmitted and received signals, motor commands and encoder signals, respectively, are communicated through a slip ring mechanism, which allows the yaw axis to move continuously without limitations.

A state space model is developed to represent the 3 DOF hover in simulations. The state space model is of the following form:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

Lyapunov functions

Lyapunov functions are commonly used to analyze dynamical systems to determine the system's stability. Furthermore, to guarantee the performance of the ETC and STC algorithms, Lyapunov functions are utilized as a measure of performance, which confirm that the de<cay rate of the closed-loop systems is satisfactory. The form of the quadratic Lyapunov function utilized in the thesis is:

$$V\big(x(t)\big) = x^T(t)Px(t)$$

The Lyapunov function can be thought of as a scalar which gives information about the energy contained in the system's current state. The controller's goal is to bring the energy contained in the system's current state towards zero.

Periodic control

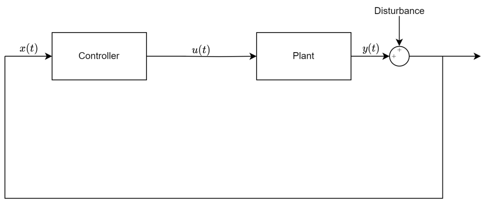
Traditional feedback control is designed with the assumption that the control action is updated every time the state of the system is sampled. All the algorithms in this thesis have been designed with the same control law:

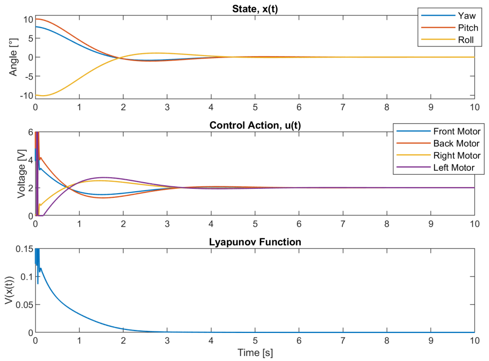
$$u(t) = Kx(t)$$

Rendering the closed-loop system:

$$\dot{x}(t) = (A + BK)x(t)$$

The gain matrix K is designed by solving a linear matrix inequality (LMI) derived from a quadratic Lyapunov function utilized in ETC and STC.





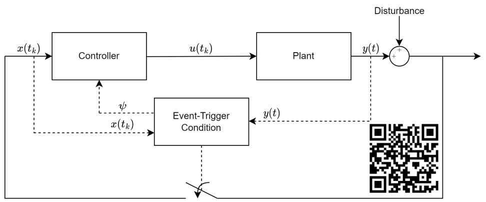
Acknowledgements

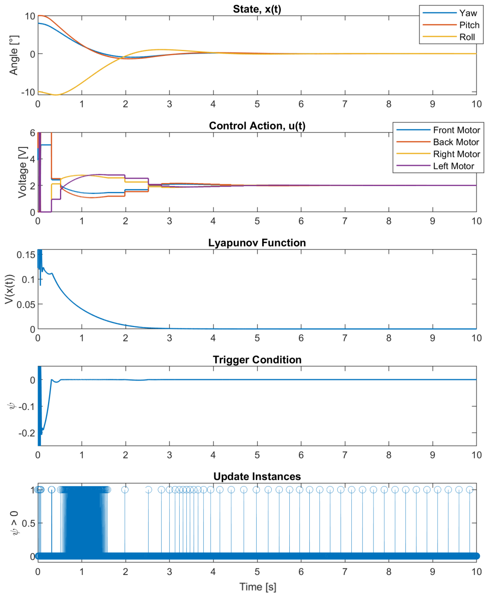
I would like to thank my supervisor, Damiano Rotondo, for outstanding help, insights and guidance during the project!

Event-triggered control

The main idea behind ETC is to update the control action aperiodically. Updates are executed when a designed triggering condition violates a specified threshold. There are different methods of designing a triggering condition, where the common factor is that the algorithm continuously monitors the system's state and compares it to the state of the system at the previous time the control action was updated. In this thesis, the triggering condition of the controller is designed based on the derivative of a quadratic Lyapunov function.

The plot below shows a simulation of ETC, and the QR code is linked to a video of ETC of the Quanser 3 DOF hover. In the plot below updates of the control action are reduced by 97,2%.

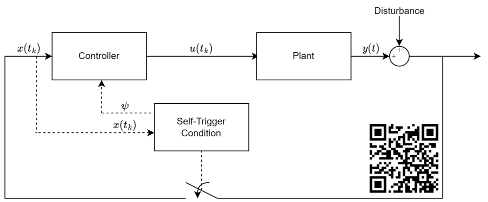


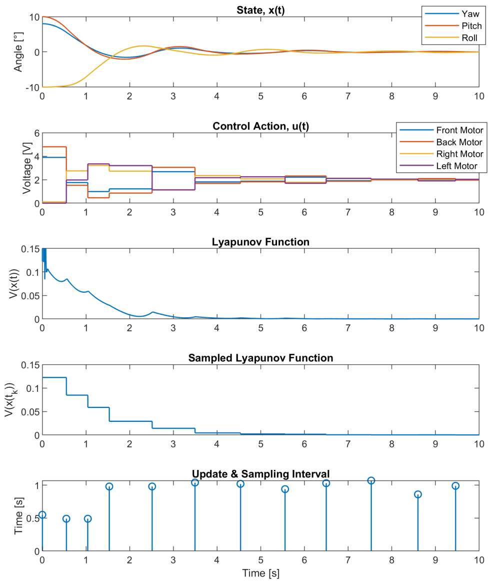


Self-triggered control

STC not only updates the control action aperiodically but sampling is also done aperiodically. The main idea behind STC is to plan the next instance at which sampling and updating are required based on a comparison between the predicted state trajectory of the system, which is estimated based on the sampled state, control input and the model of the system, and an exponentially decreasing function, which is based on the sampled state of the system. One major drawback to STC is that disturbances are not measured, hence the systems ability to deal with disturbances are quite poor.

The plot below shows a simulation of STC, and the QR code is linked to a video of STC of the Quanser 3 DOF hover. In the plot below updates of the control action are reduced by 99,88%.





Appendix B

LQR Example

The example is based on the work found in the laboratory guide on the 3 DOF hover, published by Quanser [14]. The laboratory guide presents the same weighing matrices as used here, which are presented in Equations (B.1) and (B.2). The results obtained here are found by conducting the same test as in Experiment 5 - Reference Tracking, with the difference being that the gain matrix utilised here is derived using the LQR method presented in Section 3.1 instead of the LMI based method used previously. Consider the weighing matrices:

$$Q = \begin{bmatrix} 500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 350 & 0 & 0 & 0 & 0 \\ 0 & 0 & 350 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 \end{bmatrix} \tag{B.1}$$

and

$$R = \begin{bmatrix} 0{,}01 & 0 & 0 & 0 \\ 0 & 0{,}01 & 0 & 0 \\ 0 & 0 & 0{,}01 & 0 \\ 0 & 0 & 0 & 0{,}01 \end{bmatrix} \tag{B.2}$$

Using the function `ICARE` in *MATLAB* to solve the Riccati differential Equation (3.3), the following gain matrix is computed:

$$K = \begin{bmatrix} -111{,}80 & 132{,}30 & 0 & -41{,}41 & 36{,}23 & 0 \\ -111{,}80 & -132{,}30 & 0 & -41{,}41 & -36{,}23 & 0 \\ 111{,}80 & 0 & 132{,}30 & 41{,}41 & 0 & 36{,}23 \\ 111{,}80 & 0 & -132{,}30 & 41{,}41 & 0 & -36{,}23 \end{bmatrix} \tag{B.3}$$

Table B.1 show the difference in performance for the PC strategy and ETC strategy on the 3 DOF hover, using a gain matrix designed using the LQR method and a gain matrix designed by solving the LMI in the ETC strategy.

Table B.1: IAE for each axis, the total IAE across all axes, and the UF of control action.

Strategy	Figure	Yaw	Pitch	Roll	Total	UF
PC (LQR K)	B.1	31,95	26,39	24,37	82,70	1000
ETC (LQR K)	B.2	33,15	29,16	24,18	86,49	241,33
PC (ETC K)	4.16	34,36	83,00	86,29	203,65	1000
ETC (ETC K)	4.17	25,88	78,19	73,58	177,65	97,60

From the table, it is observed that the performance in terms of the IAE metric is significantly better using the gain matrix computed using the LQR method. Meanwhile, the UF is higher using this gain matrix than using the gain matrix computed by solving the LMI in the ETC method. The following figures show that the steady-state error is relatively small compared to the steady-state error previously observed in Figures 4.16 and 4.17.

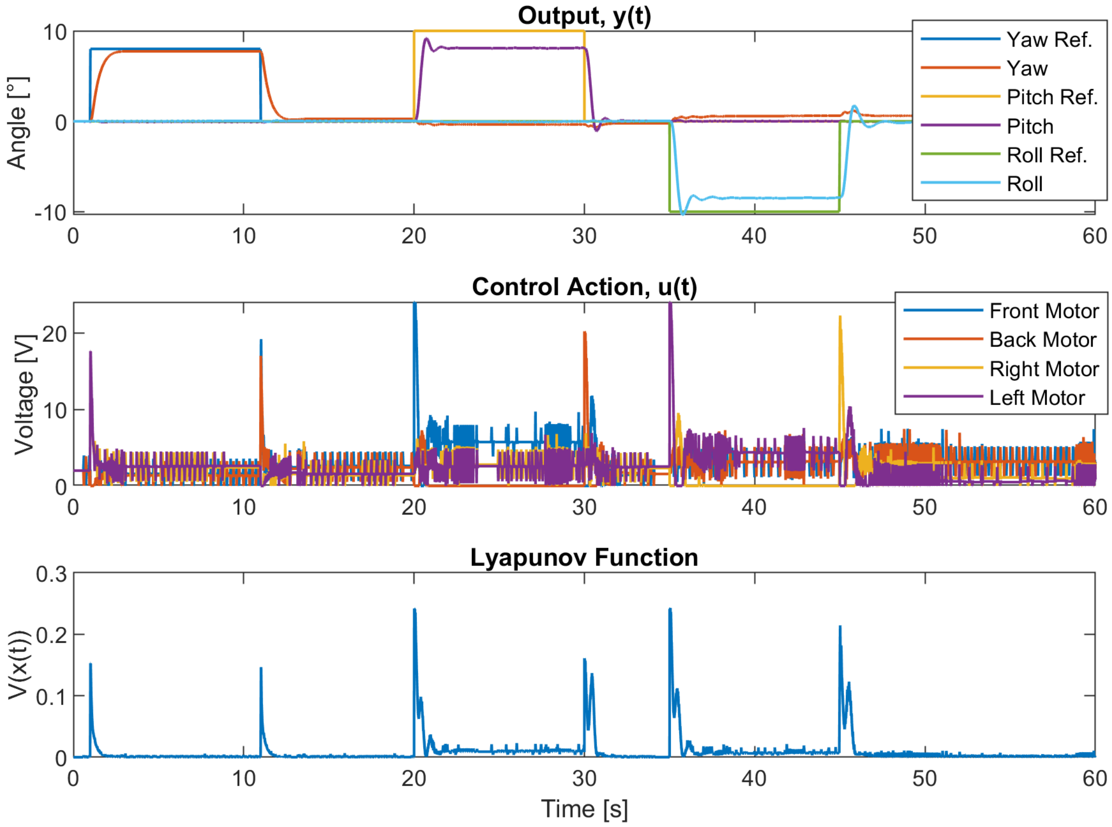


Figure B.1: On the **3 DOF hover**, the **reference tracking ability** of the **PC strategy** is tested using a gain matrix derived from solving Equation (3.2) as presented in Section 3.1.

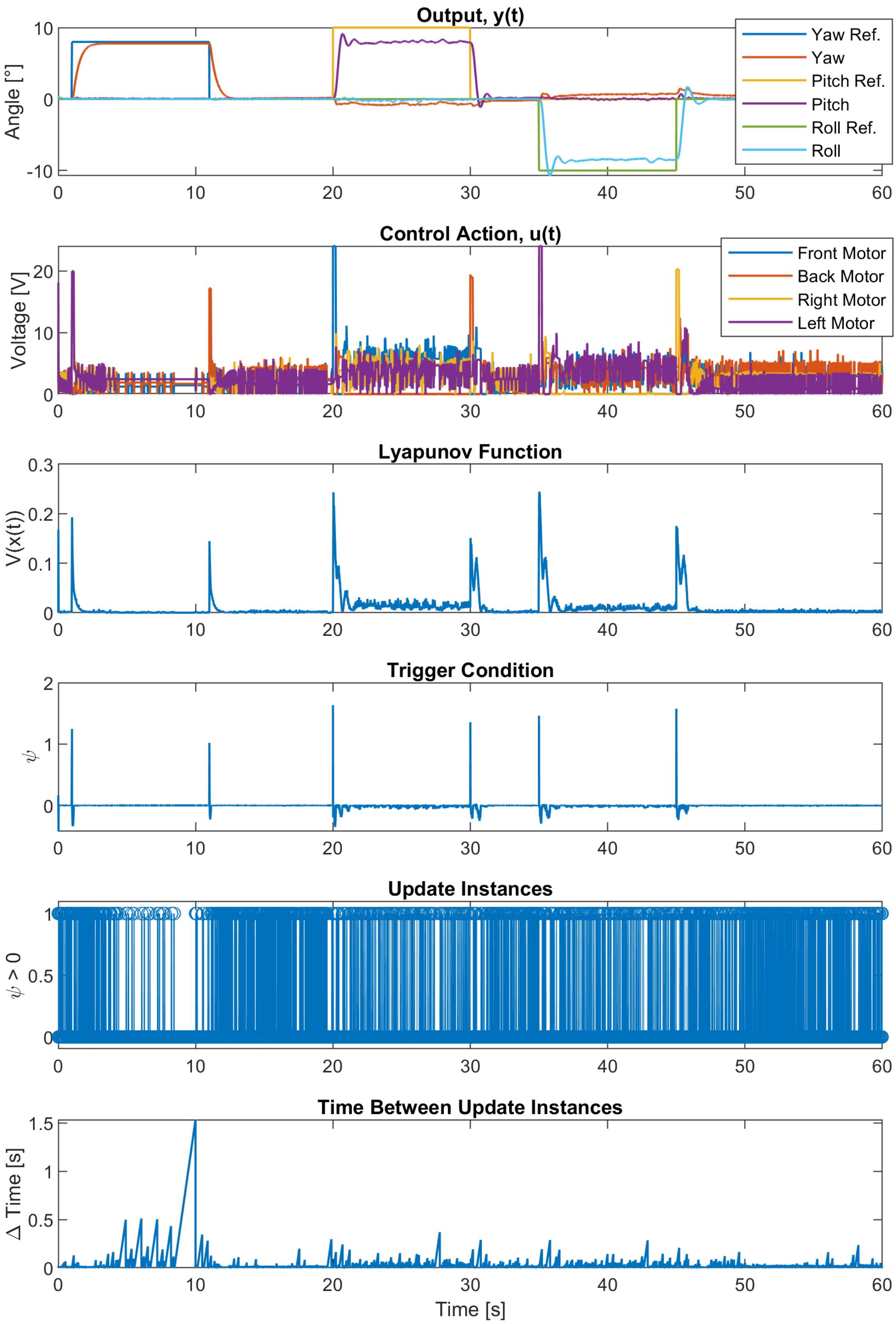


Figure B.2: On the **3 DOF hover**, the **reference tracking ability** of the **ETC strategy** is tested using a gain matrix derived from solving Equation (3.2) as presented in Section 3.1.

Appendix C

Effect of Decay Rate Parameter σ

This experiment is the same as the one presented in Experiment 3 - Initial Conditions. The reason for providing the results presented in Figure C.1 is to highlight the effect of different values of the decay rate parameter σ for the ETC strategy.

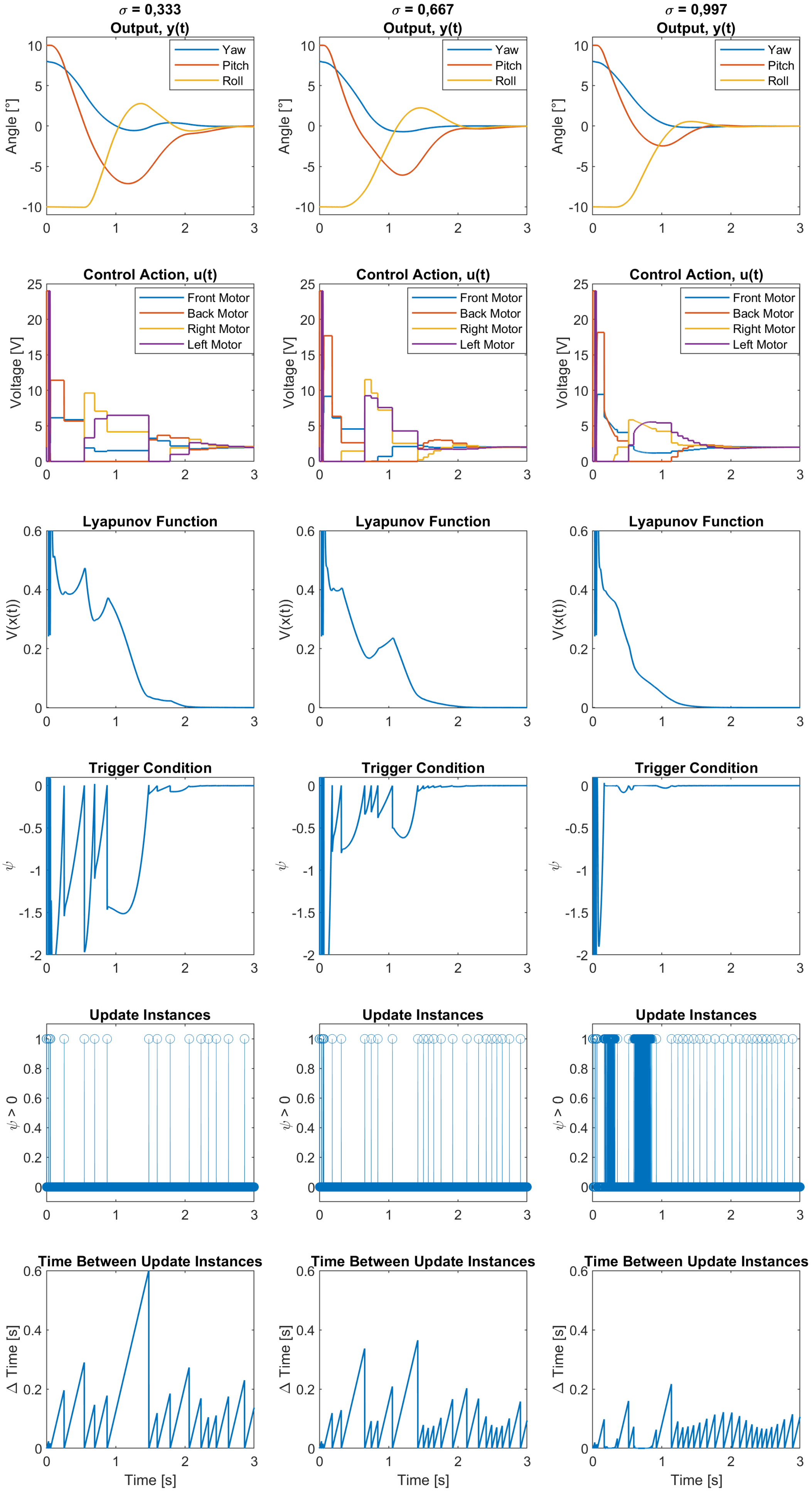


Figure C.1: Comparison of different decay rates σ of the ETC strategy.

Appendix D

ETC Experiment - With Gain Matrix Used In STC Experiments

The Appendix contains results that are used to compare the PC, ETC, and STC strategies using the same gain matrix applicable to simulations and the 3 DOF hover. The tests here are conducted the same way as Experiment 6 - Initial Conditions, using the same parameters to ensure a fair comparison between the control strategies. Figure D.1 can be compared to Figures 4.19 and 4.20. Figures D.2 and D.3 can be compared to Figure 4.21.

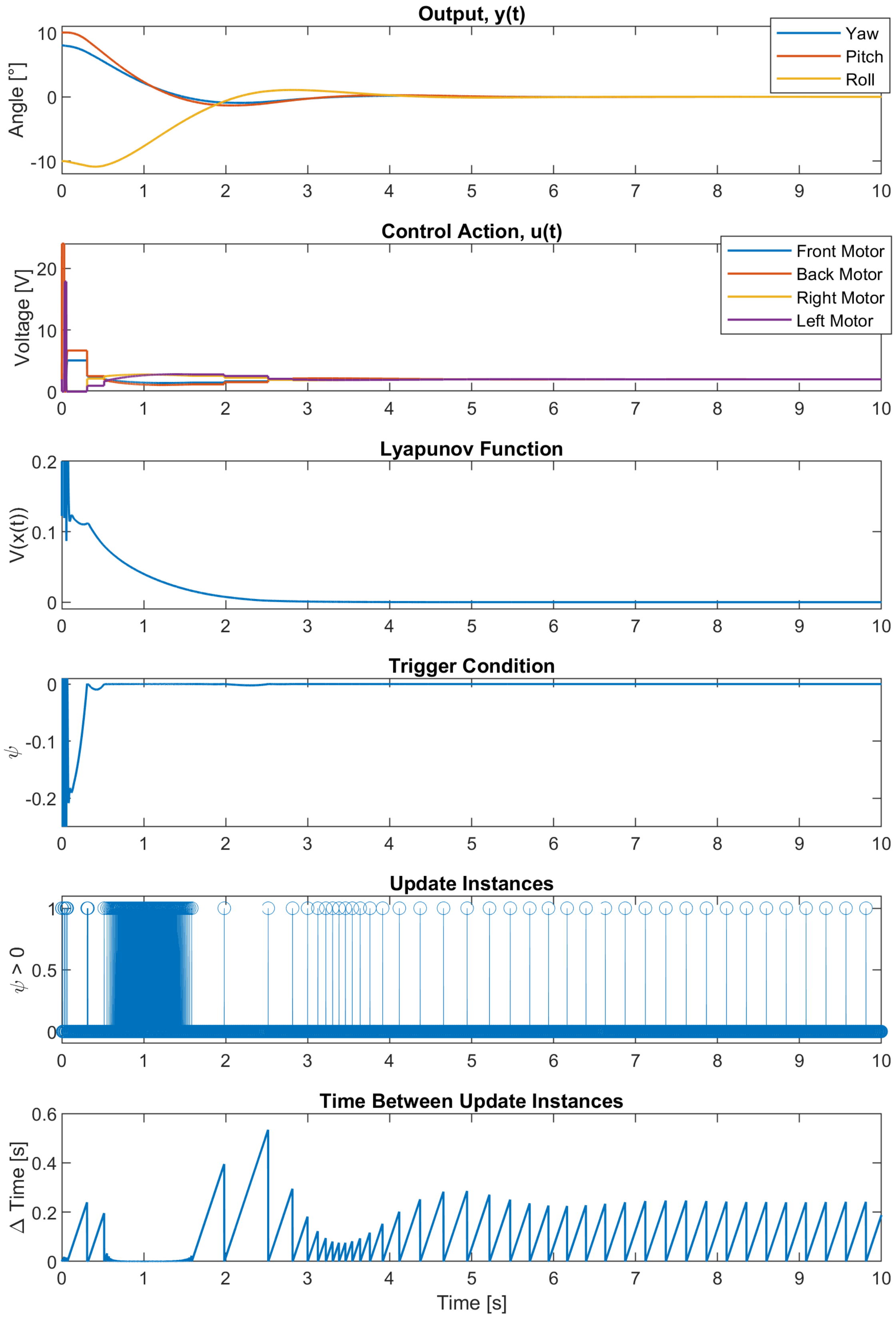


Figure D.1: In **simulation**, the **self-stabilising ability** of the **ETC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 0.4$.

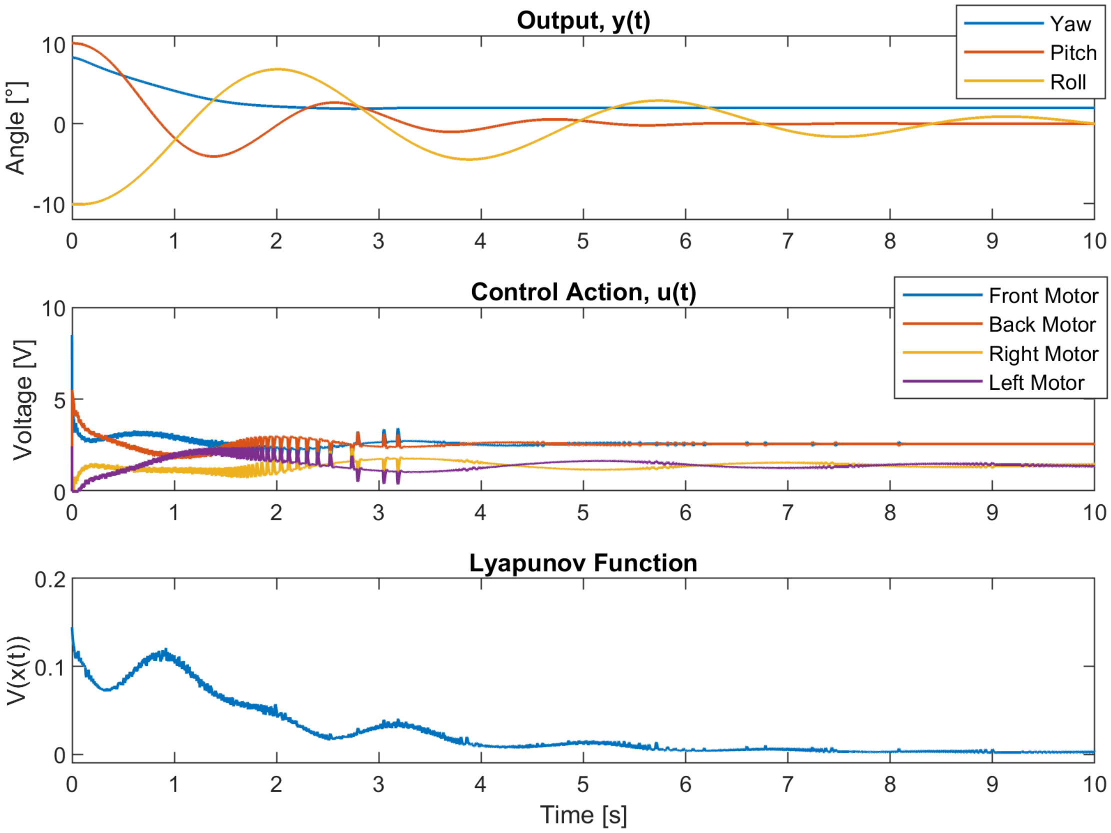


Figure D.2: On the **3 DOF hover**, the **self-stabilising ability** of the **PC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 0.4$.

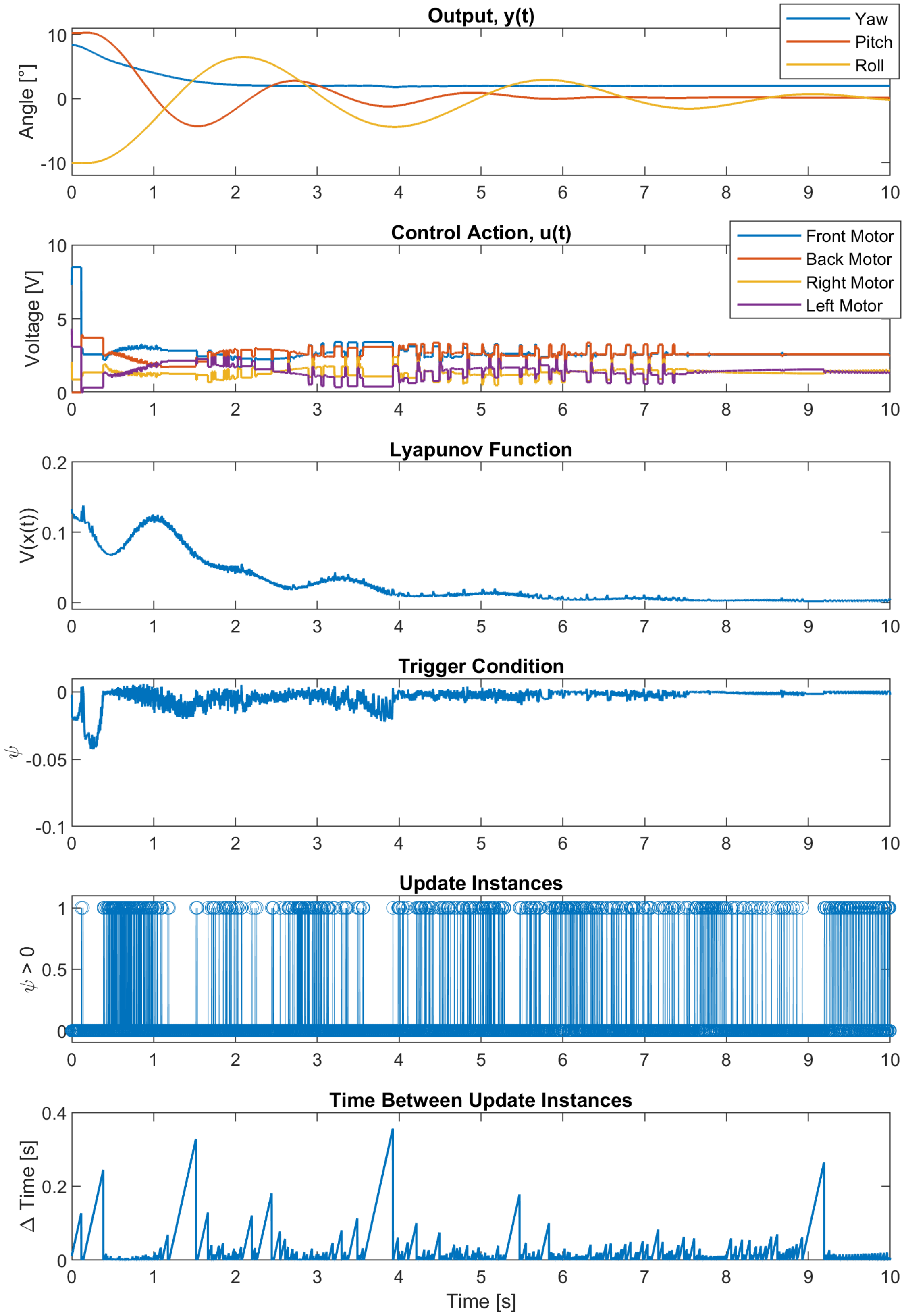


Figure D.3: On the **3 DOF hover**, the **self-stabilising ability** of the **ETC strategy** is tested using a gain matrix derived from solving the LMI presented in Equation (3.31), with $\alpha = 0.4$.

Appendix E

MATLAB Code

E.1 Solution of LMI

Listing E.1 is used to solve the LMI presented in Equation (3.20), using *YALMIP*:

```
yalmip('clear');
Y = sdpvar(4,6);      % Initializes Y - 4x6
P = sdpvar(6);        % Initializes P - 6x6
lyap_d = A*P'+B*Y+P*A'+Y'*B';
tol = 1e-10;
value = tol*eye(6);

Constraints = [P >= value, lyap_d <= -value];
ops = sdpsettings('solver', 'mosek', 'verbose', 0);
res_opt = optimize(Constraints, [], ops);

% Matrices used in the Simulink program
P = double(P);
Y = double(Y);
P = inv(P);
K = Y*P;
Q = -(A'*P+K'*B'*P + P*A+P*B*K);
```

Listing E.1: Using *MATLAB* to solve LMI, presented in Equation (3.20).

E.2 Solution of LMI, With Additional Constraint

Listing E.2 is used to solve the LMI presented in Equation (3.31), using *YALMIP*:

```matlab
yalmip('clear');
alpha = 1.8;
Y = sdpvar(4,6);     % Initializes Y - 4x6
P = sdpvar(6);       % Initializes P - 6x6
lyap_d = A*P'+B*Y+P*A'+Y'*B';
tol = 1e-10;
value = 2*alpha*P+ tol*eye(6);

% Solving the BMI
Constraints = [P >= tol*eye(6), lyap_d <= -value];
ops = sdpsettings('solver', 'mosek', 'verbose', 0);
res_opt = optimize(Constraints, [], ops);

% Matrices used in the Simulink program
P = double(P); % this is actually p*-1
Y = double(Y);
P = inv(P);
K = Y*P;
Q = -(A'*P+K'*B'*P + P*A+P*B*K);
```

Listing E.2: Using *MATLAB* to solve LMI, with added constraint, presented in Equation (3.31).

E.3 ETC - Trigger Condition

Listing E.3 presents the implementation of the triggering condition used in the ETC strategy, as presented in Equation (3.28).

```matlab
function [Lyapunov_function, triggering_condition, Trigger_Binary, flag]= fcn(x_t, x_tk, P, B, K, Q, Trigger_Binary)
sigma = 0.997;
e = x_tk - x_t;
x = x_t;

triggering_condition = (sigma-1)*x'*Q*x + 2*x'*P*B*K*e ;
Lyapunov_function = x'*P*x;
lyapunov_derivative = -x'*Q*x + 2*x'*P*B*K*e;

if triggering_condition > 0
    if Trigger_Binary == 1
        Trigger_Binary = 0;
    else
        Trigger_Binary = 1;
    end
    flag = 1;
else
    flag = 0;
end
```

Listing E.3: Implementation in *MATLAB* of the trigger condition used in the ETC strategy, as presented in Equation (3.28).

E.4 ETC Reference Tracking - Trigger Condition

Listing E.4 presents the implementation of the triggering condition used for reference tracking in the ETC strategy, as presented in Equation (3.45).

```
function [Lyapunov_function, triggering_condition,
   Trigger_Binary, flag] = fcn(x_t, x_tk, x_r, x_r_tk, P, B,
    K, Q, Trigger_Binary)
sigma = 0.997;
deltak_r = x_r_tk - x_r;
deltak_x = x_tk - x_t;
e = x_t - x_r;

triggering_condition = (sigma-1)*e'*Q*e + 2*deltak_x'*K'*B'*
   P*e - 2*deltak_r'*K'*B'*P*e;
Lyapunov_function = e'*P*e;

if triggering_condition > 0
    if Trigger_Binary == 1
        Trigger_Binary = 0;
    else
        Trigger_Binary = 1;
    end
    flag = 1;
else
    flag = 0;
end
```

Listing E.4: Implementation in *MATLAB* of the trigger condition used for reference tracking in the ETC strategy, as presented in Equation (3.45).

E.5 STC - Trigger Condition

Listing E.5 presents the implementation of the triggering condition used in the STC strategy, as presented in Equation (3.56) and the computation of the next update and sampling instance of the controller, as presented in Equation (3.55).

```
function [Lyap_func_disc, flag, Trigger_Binary, tau, t_since_update, nextSampling] = STC(x_tk, A, B, P, u_tk, Ts, lambda, Trigger_Binary, simTime, t_since_update, nextSampling)
update = false; tau = 0; integral = zeros(6);
Lyap_func_disc = (x_tk'*P*x_tk);
if simTime >= nextSampling
    h_c = 0; tau = Ts;
    while h_c <= 0
        V_exp = (x_tk'*P*x_tk)*exp(-lambda*tau);
        Traj_1 = expm(A*tau)*x_tk;
        Traj_2 = integral + Ts*0.5*(expm(-A*(tau-Ts)) + expm(-A*tau));
        integral = Traj_2;
        Traj_2 = expm(A*tau)*Traj_2*B*u_tk;
        V_traj = ((Traj_1 + Traj_2)'*P*(Traj_1 + Traj_2));
        h_c = V_traj - V_exp;
        tau = tau + Ts;
    end
    nextSampling = simTime + tau;
    update = true;
end
if update
    if Trigger_Binary == 1
        Trigger_Binary = 0;
    else
        Trigger_Binary = 1;
    end
    flag = 1;
    t_since_update =tau;
else
    flag = 0;
    t_since_update = (t_since_update - Ts);
end
end
```

Listing E.5: Implementation in *MATLAB* of the STC trigger condition and the computation of the next update and sampling instance, as presented in Equations (3.56) and (3.55).

Appendix F

SIMULINK Schemes

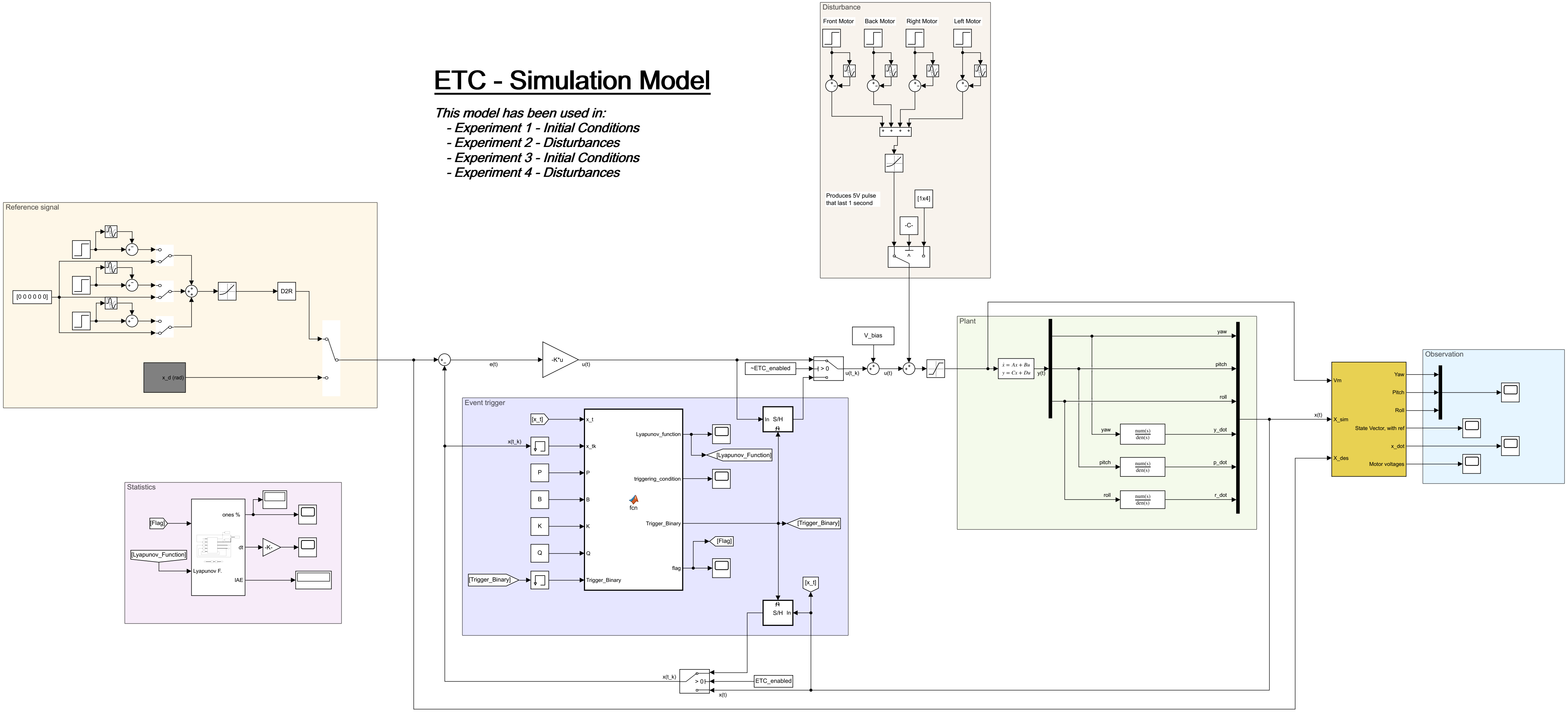
The following pages contain *SIMULINK* schemes of the following models:

1. ETC - 3 DOF hover simulation model.
2. ETC reference tracking - 3 DOF hover Simulation model.
3. ETC - 3 DOF hover laboratory model.
4. ETC reference tracking - 3 DOF hover laboratory model.
5. STC - 3 DOF hover simulation model.
6. STC - 3 DOF hover laboratory model.

ETC - Simulation Model

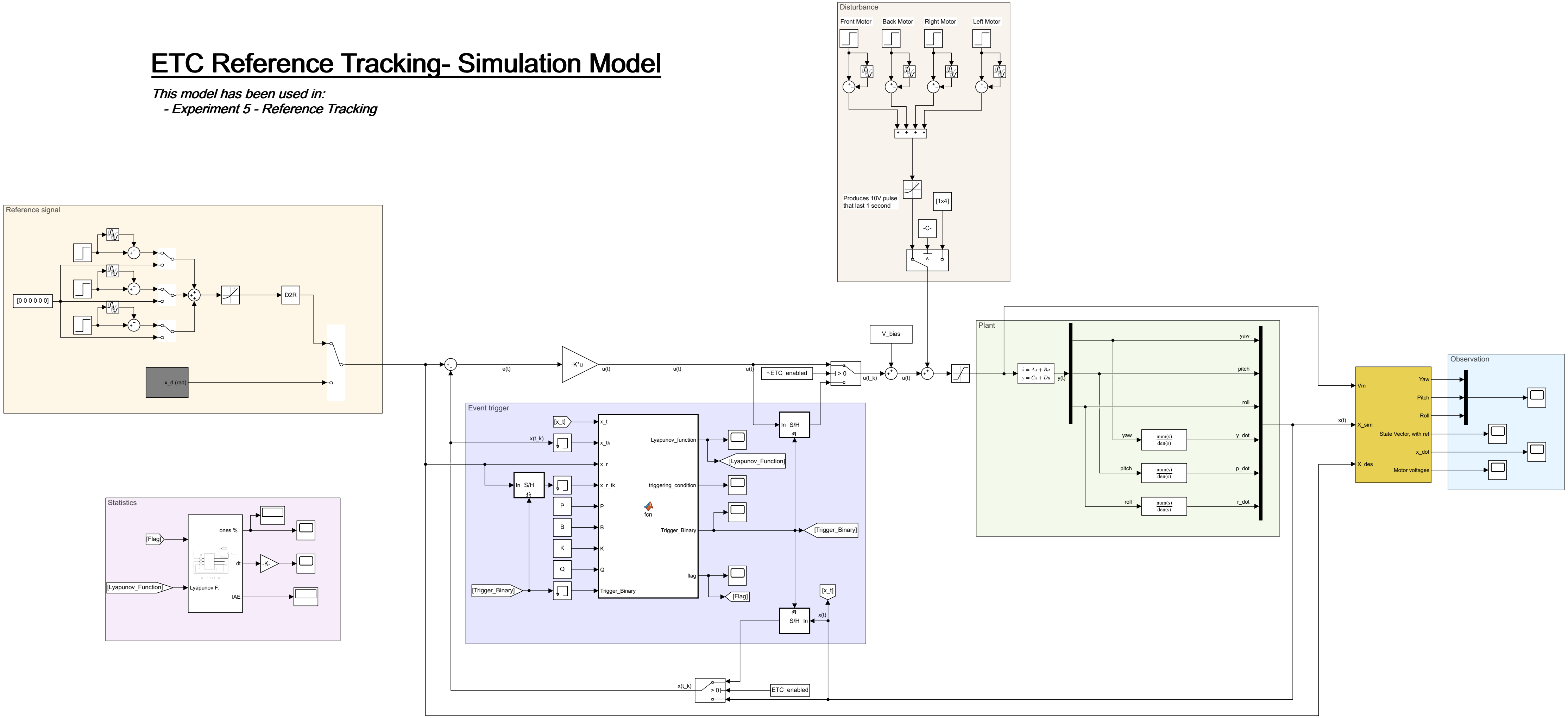
This model has been used in:

- *Experiment 1 - Initial Conditions*
- *Experiment 2 - Disturbances*
- *Experiment 3 - Initial Conditions*
- *Experiment 4 - Disturbances*



ETC Reference Tracking- Simulation Model

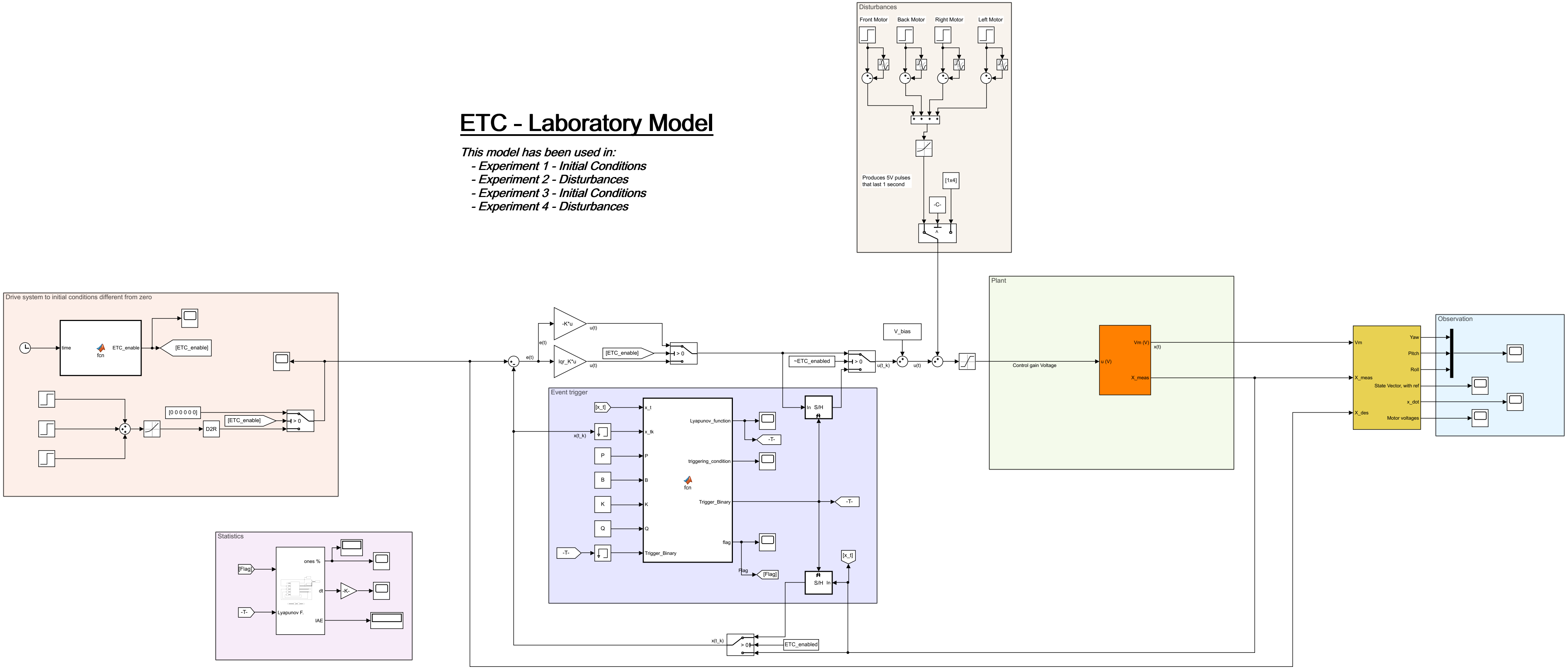
This model has been used in:
- Experiment 5 - Reference Tracking



ETC - Laboratory Model

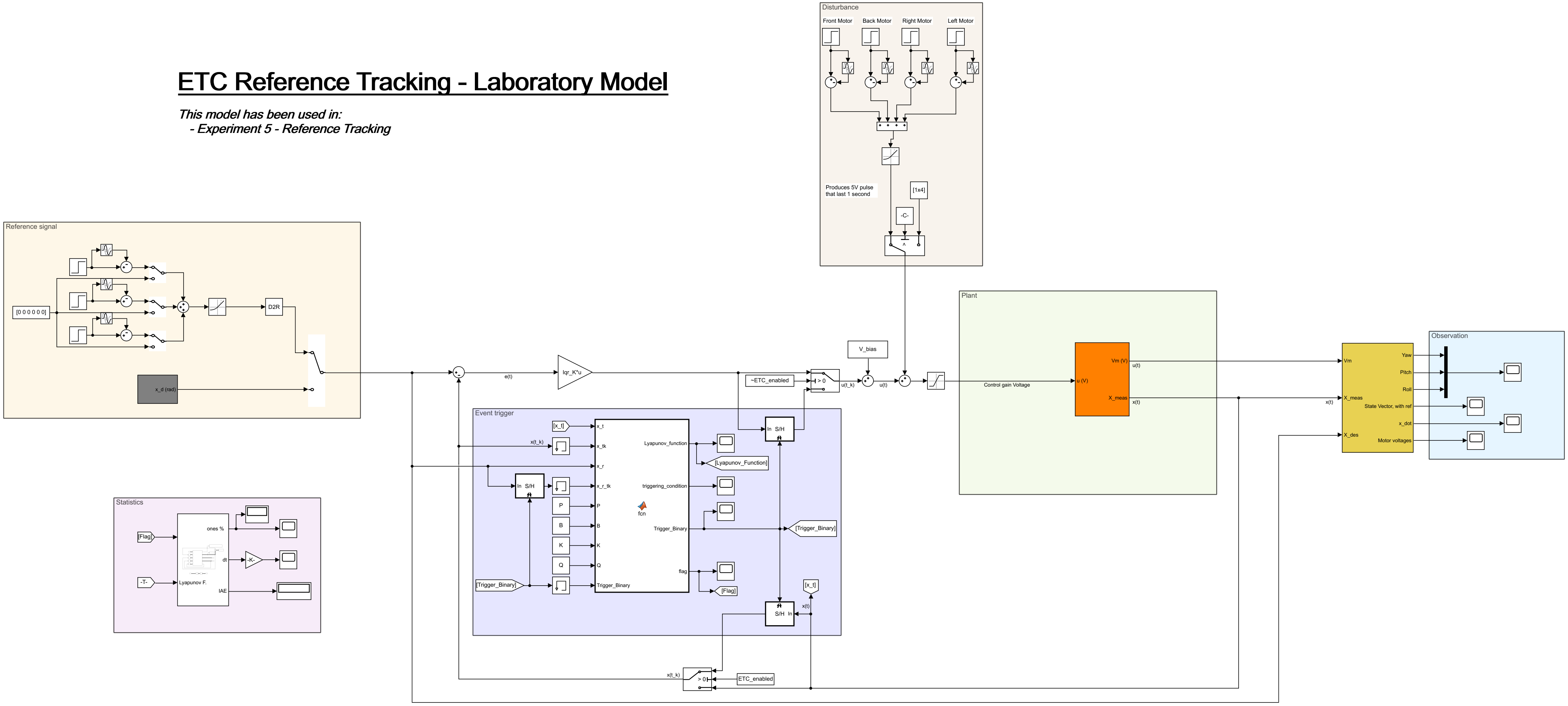
This model has been used in:

- *Experiment 1 - Initial Conditions*
- *Experiment 2 - Disturbances*
- *Experiment 3 - Initial Conditions*
- *Experiment 4 - Disturbances*



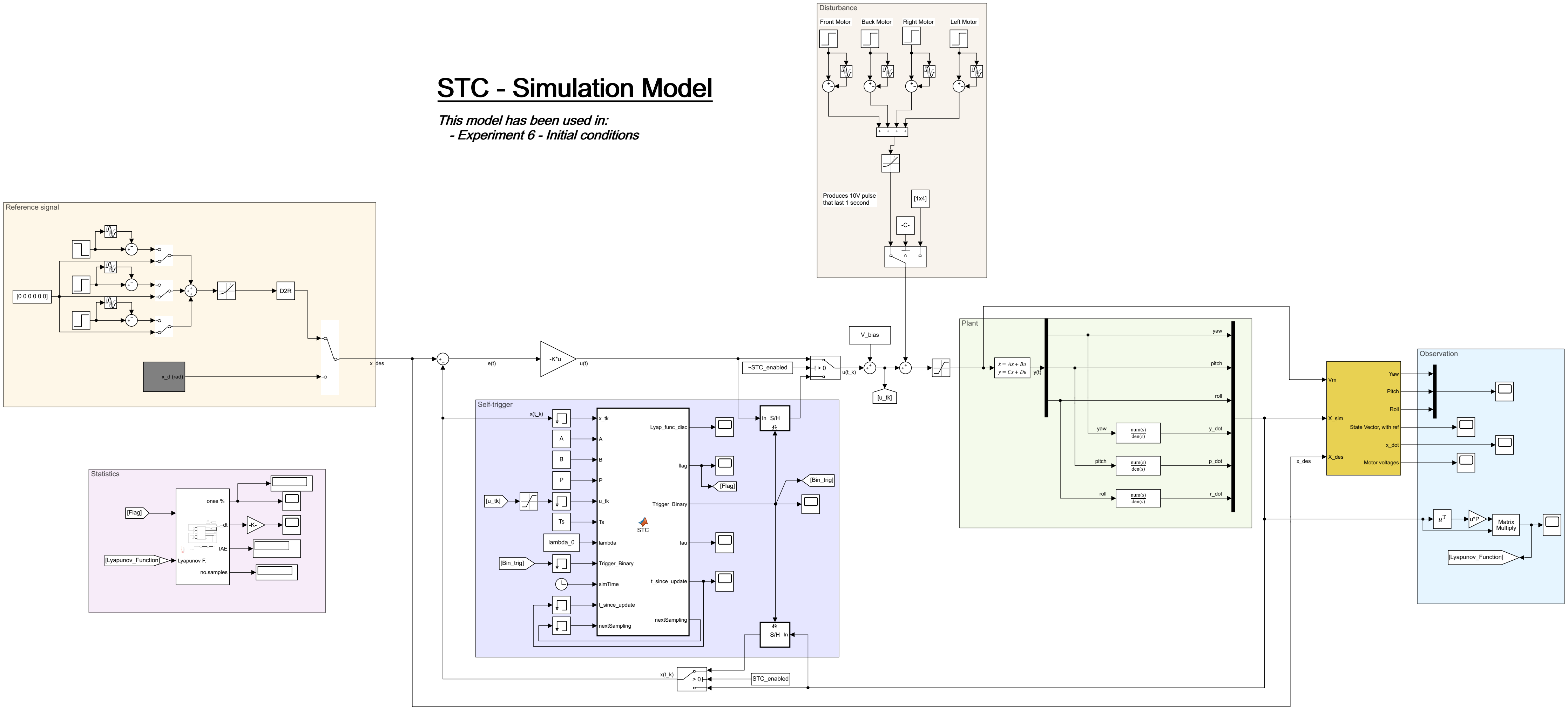
ETC Reference Tracking - Laboratory Model

This model has been used in:
- Experiment 5 - Reference Tracking



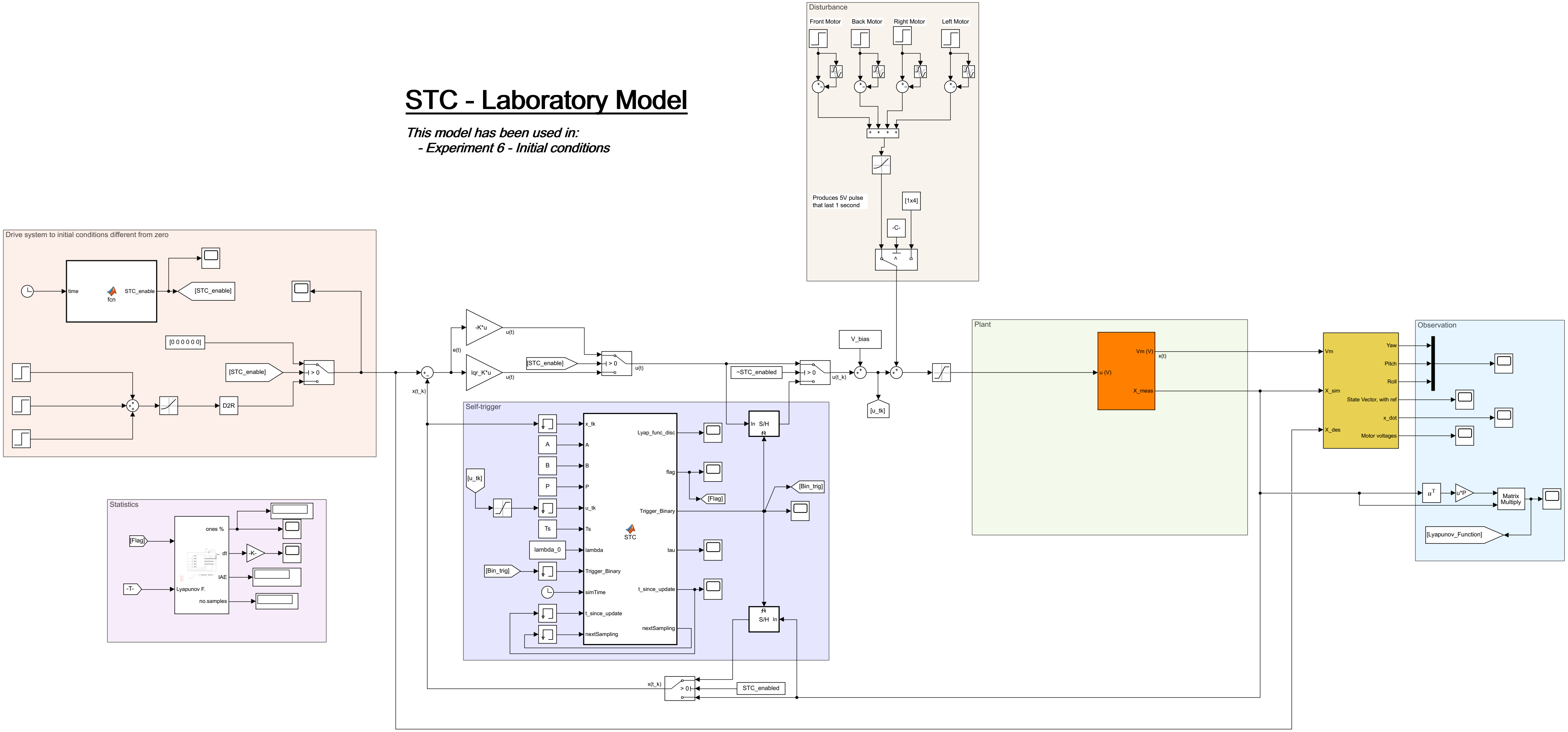
STC - Simulation Model

This model has been used in:
- Experiment 6 - Initial conditions



STC - Laboratory Model

This model has been used in:
- Experiment 6 - Initial conditions



Bibliography

[1] Gene F. Franklin, J. Da Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall Press, USA, 7th edition, 2014. ISBN 0133496597.

[2] 3 dof hover - flight dynamics and control of vertical lift-off vehicles. Quanser, 2023. URL https://www.quanser.com/products/3-dof-hover/. Accessed 24 January 2023.

[3] 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.

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

[6] Karl Johan Åström and Bo Bernhardsson. Comparison of periodic and event based sampling for first-order stochastic systems. *IFAC Proceedings Volumes*, 32(2):5006–5011, 1999. doi: https://doi.org/10.1016/S1474-6670(17)56852-4.

[7] Karl-Erik Åarzén. A simple event-based pid controller. *IFAC Proceedings Volumes*, 32(2):8687–8692, 1999. doi: https://doi.org/10.1016/S1474-6670(17)57482-0.

[8] Manel Velasco, Josep Fuertes, and Pau Marti. The self triggered task model for real-time control systems. In *Work-in-Progress Session of the 24th IEEE Real-Time Systems Symposium (RTSS03)*, volume 384, pages 67–70. Citeseer, 2003.

[9] Paulo Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007. doi: 10.1109/TAC.2007.904277.

[10] W.P.M.H. Heemels, K.H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 3270–3285, 2012. doi: 10.1109/CDC.2012.6425820.

[11] Yul Y. Nazaruddin, Augie Widyotriatmo, Tua A. Tamba, Muhammad S. Arifin, and Rival A. Santosa. Communication-efficient optimal-based control of a quadrotor uav by event-triggered mechanism. In *2018 5th Asian Conference on Defense Technology (ACDT)*, pages 96–101, 2018. doi: 10.1109/ACDT.2018.8593118.

[12] J. Téllez-Guzmán, Fermi Castellanos, Sylvain Durand, Nicolas Marchand, and R. Lucio-Maya. Event-based lqr control for attitude stabilization of a quadrotor. 10 2012.

[13] J. F. Guerrero-Castellanos, J. J. Téllez-Guzmán, S. Durand, N. Marchand, and J. U. Alvarez-Muñoz. Event-triggered nonlinear control for attitude stabilization of a quadrotor. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 584–591, 2013. doi: 10.1109/ICUAS.2013.6564736.

[14] J.Apkarian and M.Lévis. Laboratory guide, 3 dof hover experiment for matlab /simulink users. Quanser, 2013. Accessed 24 January 2023.

[15] William L Brogan. Modern control theory, 1991.

[16] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control.* Cambridge University Press, 2022.

[17] Robert L Williams, Douglas A Lawrence, et al. *Linear state-space control systems.* John Wiley & Sons, 2007.

[18] Arieh Iserles. *A first course in the numerical analysis of differential equations.* Number 44. Cambridge university press, 2009.

[19] Kendall Atkinson. *An introduction to numerical analysis.* John wiley & sons, 1991.

[20] Guang-Ren Duan and Hai-Hua Yu. *LMIs in control systems: analysis, design and applications.* CRC press, 2013.

[21] J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[22] W. P. M. H. Heemels, M. C. F. Donkers, and Andrew R. Teel. Periodic event-triggered control for linear systems. *IEEE Transactions on Automatic Control*, 58 (4):847–861, 2013. doi: 10.1109/TAC.2012.2220443.

[23] S Boyd, L El Ghaoui, E Feron, and V Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15. SIAM, 1994.

[24] Edward John Routh. *A treatise on the stability of a given state of motion, particularly steady motion: being the essay to which the Adams prize was adjudged in 1877, in the University of Cambridge.* Macmillan and Company, 1877.

[25] Fairouz Zobiri. *On the design of event-and self-triggered controllers for certain classes of dynamical systems.* PhD thesis, Université Grenoble Alpes, 2019.

[26] Manuel Mazo, Adolfo Anta, and Paulo Tabuada. On self-triggered control for linear systems: Guarantees and complexity. In *2009 European Control Conference (ECC)*, pages 3767–3772. IEEE, 2009.

[27] Manuel Mazo, Adolfo Anta, and Paulo Tabuada. An iss self-triggered implementation of linear controllers. *Automatica*, 46(8):1310–1314, 2010. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2010.05.009.

[28] João Almeida, Carlos Silvestre, and António M. Pascoal. Self-triggered output feedback control of linear plants. In *Proceedings of the 2011 American Control Conference*, pages 2831–2836, 2011. doi: 10.1109/ACC.2011.5991325.

[29] K David Young, Vadim I Utkin, and Umit Ozguner. A control engineer's guide to sliding mode control. *IEEE transactions on control systems technology*, 7(3): 328–342, 1999.

[30] MCF Donkers and WPMH Heemels. Output-based event-triggered control with guaranteed $\mathcal{L}_\infty$-gain and improved and decentralized event-triggering. *IEEE Transactions on Automatic Control*, 57(6):1362–1376, 2011.