

# MODELLING, CONTROL AND CONSTRUCTION OF TRICOPTER UNMANNED AERIAL VEHICLES

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN THE FACULTY OF SCIENCE AND ENGINEERING

2022

**Daniel N. Abara**

School of Engineering, Department of Electrical and Electronic Engineering

# Contents

<b>Abstract</b>	<b>12</b>
<b>Declaration</b>	<b>13</b>
<b>Copyright Statement</b>	<b>14</b>
<b>Dedication</b>	<b>15</b>
<b>Acknowledgements</b>	<b>16</b>
<b>Publications</b>	<b>17</b>
<b>Nomenclature</b>	<b>18</b>
<b>1 Introduction</b>	<b>22</b>
1.1 Background and Motivation . . . . .	22
1.1.1 Multi-agent Systems . . . . .	27
1.2 Contributions to the state-of-the-art . . . . .	30
1.3 Thesis Organization . . . . .	31
<b>2 Preliminaries</b>	<b>34</b>
2.1 Linear Algebra . . . . .	34
2.1.1 Linear subspaces . . . . .	34
2.1.2 Symmetric and semidefinite matrices . . . . .	35
2.1.3 Cross products . . . . .	35
2.1.4 Kronecker products . . . . .	36
2.1.5 Eigenvalues and Eigenvectors . . . . .	37
2.2 Linear Systems Theory . . . . .	37



2.2.1	Description of systems . . . . .	37
2.2.2	Stability, Controllability and Observability . . . . .	38
2.2.3	State space realizations . . . . .	39
2.2.4	Internal stability . . . . .	39
2.3	Graph theory . . . . .	40
2.4	Homogeneous multi-agent systems . . . . .	41
2.5	Negative Imaginary Systems . . . . .	42
2.5.1	Properties of networked NI and SNI systems . . . . .	42
2.5.2	Eigenvalue loci theory . . . . .	43
2.5.3	Properties of eigenvalue loci of networked NI (SNI) systems . . . . .	44
2.6	Quaternion Maths . . . . .	45
2.7	Coordinate Rotations and Rotation matrices . . . . .	46
2.7.1	Rotation matrices . . . . .	48
2.7.2	Quaternion rotation operators . . . . .	48
2.7.3	Euler angles and rotation sequences . . . . .	49
2.7.4	Relationship between Quaternions and Euler angles . . . . .	51
2.8	Summary . . . . .	51
<b>3</b>	<b>Development and stabilization of a low-cost single-tilt tricopter</b>	<b>52</b>
3.1	Introduction . . . . .	52
3.2	Mathematical Modeling . . . . .	53
3.2.1	Forces and Torques . . . . .	54
3.2.2	Newton-Euler Model . . . . .	56
3.3	Model Parameters . . . . .	57
3.3.1	Moments of Inertia . . . . .	57
3.3.2	Thrust and Torque Constants . . . . .	58
3.4	Hover Control design . . . . .	60
3.4.1	Linearised Model . . . . .	60
3.4.2	Control Allocation . . . . .	60
3.4.3	PID Cascade scheme . . . . .	61
3.5	Simulation Results . . . . .	63
3.6	Experimental Results . . . . .	66

3.6.1	Platform Description . . . . .	66
3.6.2	Hover flight test . . . . .	66
3.7	Summary . . . . .	68
<b>4</b>	<b>Development and control of Multirotor-tilting tricopter with direct actuation for position control</b>	<b>69</b>
4.1	Introduction . . . . .	69
4.2	Modelling of the multirotor-tilting tricopter . . . . .	71
4.2.1	Tricopter Forces and Moments . . . . .	71
4.2.2	Dynamic model . . . . .	75
4.3	Parameter Identification . . . . .	75
4.3.1	Moment of Inertia . . . . .	76
4.3.2	Thrust and Torque Constants . . . . .	78
4.4	Attitude control using Quaternion feedback control (QFB) technique .	79
4.4.1	QFB control scheme . . . . .	79
4.4.2	Quaternion error . . . . .	80
4.5	Attitude control using MPC . . . . .	81
4.5.1	Quasi-LPV modelling of the tricopter . . . . .	81
4.5.2	MPC formulation . . . . .	82
4.6	Matlab simulation results for attitude control . . . . .	85
4.6.1	Simulation results using QFB technique . . . . .	85
4.6.2	Comparative simulation study between QFB and MPC . . . . .	87
4.7	6-DOF control of tricopter . . . . .	91
4.8	Experiments . . . . .	95
4.8.1	Platform Description . . . . .	96
4.8.2	Mixer for position control . . . . .	97
4.8.3	Mixer Implementation . . . . .	98
4.8.4	Experimental Results . . . . .	99
4.9	Summary . . . . .	101
<b>5</b>	<b>A negative imaginary robust formation control scheme for a group of networked tricopters over inner-loop sliding-mode control</b>	<b>103</b>
5.1	Introduction . . . . .	103

5.1.1	Problem Formulation . . . . .	107
5.2	Multi-tilt tricopter modelling . . . . .	107
5.2.1	Linearisation using Sliding Mode Control . . . . .	107
5.3	System Identification of NI systems . . . . .	111
5.3.1	Continuous time closed-loop system identification . . . . .	111
5.3.2	System identification of a tricopter enforcing NI property . . . .	113
5.4	Formation control of multi-tilt tricopters . . . . .	116
5.4.1	Closed-loop stability of networked NI/SNI system with ‘mixed’ SNI+VSP controller . . . . .	116
5.4.2	Formation control of networked multi-tilt tricopters using ‘mixed’ SNI+VSP controller . . . . .	120
5.5	Case study and simulation results . . . . .	122
5.5.1	Formation control of a group of six tricopters . . . . .	122
5.6	Summary . . . . .	129
<b>6</b>	<b>Concluding Remarks</b>	<b>130</b>
6.1	Contributions . . . . .	130
6.2	Directions for Future Research . . . . .	132
	<b>Bibliography</b>	<b>134</b>
<b>A</b>	<b>Platform Description</b>	<b>147</b>
A.1	Hardware Design . . . . .	147
A.1.1	Pixhawk Flight Controller . . . . .	148
A.1.2	Additional Sensors . . . . .	149
A.1.3	Propulsion and Power System . . . . .	150
A.1.4	Heading control concept . . . . .	151
A.2	Software Description . . . . .	153
A.2.1	PX4 Architecture . . . . .	153
A.2.2	QGroundControl Interface . . . . .	153
A.3	Summary . . . . .	154
<b>B</b>	<b>Tricopter System Specification</b>	<b>155</b>
B.1	Single-tilt tricopter . . . . .	155

B.1.1	Identification . . . . .	155
B.1.2	Performance Summary . . . . .	155
B.1.3	System Hardware Description . . . . .	156
B.2	Multi-tilt tricopter . . . . .	158
B.2.1	Identification . . . . .	158
B.2.2	Performance Summary . . . . .	158
B.2.3	System Hardware Description . . . . .	158
<b>C</b>	<b>MPC Formulation</b>	<b>161</b>
C.1	Forward Euler method of discretization . . . . .	161
C.2	Cost function simplification . . . . .	162
C.2.1	Part I . . . . .	162
C.2.2	Part II - Future State Prediction Formula . . . . .	163
C.2.3	Part III . . . . .	164
<b>D</b>	<b>Row Reduction using Gaussian Elimination</b>	<b>166</b>
<b>E</b>	<b>Mixer Implementation in Pixhawk</b>	<b>169</b>
E.1	Airframe Configuration file . . . . .	169
E.2	Multi-tri Geometry file . . . . .	171
E.3	MAIN Mixer file - mt_tri_y_yaw-.main.mix . . . . .	172
E.4	AUX Mixer file - mt_tri_y_yaw-.aux.mix . . . . .	172

Word count: 32,300

# List of Tables

3.1	Summary of estimated parameters . . . . .	59
3.2	PID gains for attitude rate loop . . . . .	62
3.3	PID gains for attitude loop . . . . .	62
3.4	PID gains for velocity loop . . . . .	62
3.5	PID gains for position loop . . . . .	63
4.1	Trifilar experiment: Time for 10 oscillations. Ax means axis hence $t_x$ is the measured time for 10 oscillations around x axis, $h_z$ is the measured time for 10 oscillations of the empty hoop around the z-axis. All measurements are in seconds. . . . .	79
5.1	Summary of tricopter parameters . . . . .	110

# List of Figures

1.1	Tricopter Configurations [images from Google]. . . . .	23
(a)	T-shaped tricopter airframe . . . . .	23
(b)	Y-shaped Tricopter schematic . . . . .	23
(c)	Y-shaped Tricopter schematic . . . . .	23
2.1	Positive Feedback interconnection of two systems $\Delta(s)$ and $M(s)$ . . .	39
2.2	Network graphs with different topologies. . . . .	40
(a)	Undirected graph . . . . .	40
(b)	Directed graph . . . . .	40
2.3	Nyquist $\mathcal{D}$ -contour in the $s$ -plane, where $\omega = 0_+, 0_-$ and $\omega = +\infty, -\infty$ are denoted by $j0_+, j0_-$ and $+j\infty, -j\infty$ respectively, $\Omega_0 = \{s \mid s = \varepsilon e^{j\theta}, \varepsilon \in \mathbb{R}_{>0}, \varepsilon \rightarrow 0, -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}\}$ , $\Omega_R = \{s \mid s = Re^{j\theta}, R \in \mathbb{R}_{>0}, R \rightarrow +\infty, -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}\}$ and $\sigma \in (0, \sigma^*)$ is a finite range. . . . .	44
2.4	Rotation of coordinates . . . . .	47
3.1	Forces and torques acting on single-tilt tricopter and coordinate systems.	54
3.2	Top view diagram of tricopter where $l_0$ is the length of each arm, $l_1 = l_0 \sin \frac{\pi}{3}, l_2 = l_0 \cos \frac{\pi}{3}$ . . . . .	54
3.3	Derivation of moments of inertia of a tricopter. Note that $l = l_0$ . . . . .	57
3.4	Freebody diagram of thrust stand . . . . .	58
3.5	Measurement of thrust and torque constants. . . . .	58
3.6	Thrust and drag-torque constants data with regression fit. . . . .	59
3.7	Cascaded-PID control architecture. . . . .	62
3.8	$x, y$ and $z$ (altitude) position of tricopter in inertial coordinates. . . . .	63
3.9	Attitude of tricopter. . . . .	64

3.10	Control inputs and rotor speeds of the tricopter. . . . .	65
3.11	Side view of the single-tilt tricopter assembly. . . . .	66
3.12	Attitude and PWM commands from test. . . . .	67
4.1	Forces and torques acting on tricopter and coordinate systems . . . . .	72
4.2	Top view of the tricopter showing all frames where CW and CCW mean clockwise and counter clockwise respectively. . . . .	72
4.3	Trifilar experiment with tricopter set for measuring $J_z$ . . . . .	76
4.4	Trifilar experiment with tricopter set for measuring $J_x$ and $J_y$ . . . . .	77
4.5	Thrust and drag-torque constants data with 93.82% fit for $k_t$ and 96.78% fit for $k_d$ . . . . .	78
4.6	Quaternion feedback control scheme. . . . .	79
4.7	LPV-MPC scheme . . . . .	82
4.8	Attitude response to demanded heading of -0.12 rad ( $-7$ deg.) after 3s, $([\phi = 0, \theta = 0, \psi = -0.12]^\top \Rightarrow [q_0 = 1, q_1 = 0, q_2 = 0, q_3 = -0.059]^\top)$ . . . . .	86
4.9	Attitude response in Euler angles $\boldsymbol{\eta}$ , $([\phi = 0, \theta = 0, \psi = -0.12]^\top \Rightarrow$ $[q_0 = 1, q_1 = 0, q_2 = 0, q_3 = -0.059]^\top)$ . . . . .	87
4.10	Quaternion error $\mathbf{q}_e$ and Angular velocity $\boldsymbol{\omega}^b$ . . . . .	87
4.11	Control torques and actuator outputs, $\boldsymbol{\tau}_m^b$ , $\alpha_i$ and $\omega_i$ stand for torques, servo angles and motor speeds respectively. . . . .	88
4.12	Attitude response to demanded heading of -0.14 rad with QFB and MPC . . . . .	89
4.13	Comparing control torques between QFB and MPC schemes . . . . .	89
4.14	Comparing motor speeds between QFB and MPC schemes . . . . .	90
4.15	Comparing servo angles between QFB and MPC schemes . . . . .	91
4.16	Position of tricopter using translational forces only without changing attitude . . . . .	93
4.17	Tricopter response to rectangular trajectory with motion achieved using translational control forces only. . . . .	93
4.18	Tilt angles applied to the servo motors to enable motions in $x$ or $y$ directions. . . . .	94
4.19	Attitude response to reference $\mathbf{q}_d = [1 \ 0 \ 0 \ 0]^\top$ ( $\boldsymbol{\eta} = [0 \ 0 \ 0]^\top$ ) with position control for 6-DOF control. . . . .	95
4.20	Tricopter platform on test stand . . . . .	95

4.21	Tilting mechanism assembly. . . . .	96
4.22	Thrust vectoring via direct actuation . . . . .	97
4.23	Pseudo-code for AUX mixer implementation . . . . .	99
4.24	Servo outputs when using direct actuation for $x$ motion . . . . .	100
4.25	Servo outputs when using direct actuation for $y$ motion . . . . .	101
5.1	SMC control scheme for linearising tricopter dynamics . . . . .	109
5.2	Tricopter response to spiral trajectory with SMC . . . . .	110
5.3	Tricopter attitude with SMC . . . . .	111
5.4	Frequency response comparison: $x$ and $y$ channels . . . . .	114
5.5	Frequency response comparison: $z$ and roll channels . . . . .	115
5.6	Frequency response comparison: pitch and yaw channels . . . . .	115
5.7	A formation control scheme for a group of networked SMC-linearized tricopter agents (being SNI) involving a decoupled ‘mixed’ SNI+VSP controller $K(s)$ with $K(0) > 0$ . . . . .	116
5.8	Closed-loop stability of a negative feedback interconnection containing a networked ‘mixed’ SNI+VSP system $K(s)$ cascaded with a decentralised SNI system $M(s)$ . . . . .	117
5.9	(a) All the eigenvalue loci $\rho_i(j\omega)$ of $(\mathcal{L}+G) \otimes K(s)M(s)$ remain confined within the Green coloured region $\forall \omega \in \mathbb{R} \cup \{\infty\}$ when $M(s)$ is SNI with $M(0) > 0$ and $K(s)$ is ‘mixed’ SNI+VSP with $K(0) > 0$ ; (b) Nyquist $\mathcal{D}$ -contour in the $s$ -plane. . . . .	118
	(a) . . . . .	118
	(b) . . . . .	118
5.10	An equivalent block diagram of the formation control scheme shown in Figure 5.7 for SMC-linearised networked tricopter agents. Note $\mathcal{L}_G = (\mathcal{L} + G)$ . . . . .	121
5.11	Undirected network interaction topology. . . . .	124
5.12	Group formation of six tricopter agents with SNI+VSP controllers with formation configuration switched from diamond to triangle after 10 seconds. . . . .	125
5.13	$x$ position responses. . . . .	126
5.14	$y$ position responses. . . . .	126



5.15	altitude $z$ responses. . . . .	127
5.16	roll attitude responses. . . . .	127
5.17	pitch attitude responses. . . . .	128
5.18	yaw attitude responses. . . . .	128
A.1	Block diagram of the setup. . . . .	147
A.2	Pixhawk 1 Flight controller . . . . .	148
A.3	PX4Flow optical flow sensor. . . . .	149
A.4	Lidar-Lite V3 range-finder. . . . .	150
A.5	Main Power supply components . . . . .	151
	(a) Matek PDB . . . . .	151
	(b) LiPo pack . . . . .	151
A.6	Heading control; CW and CCW mean Clockwise and Counter clockwise respectively. . . . .	151
	(a) heading control operation . . . . .	151
	(b) tilt action . . . . .	151
A.7	Single-tilt tricopter Assembly . . . . .	152
A.8	Multi-tilt tricopter Assembly . . . . .	152
A.9	QGroundControl Ground station application. . . . .	154

# The University of Manchester

**Daniel N. Abara**

**Doctor of Philosophy**

**Modelling, Control and Construction of Tricopter Unmanned Aerial Vehicles**

**May 24, 2022**

This thesis deals with the development and control of low-cost single-tilt and multi-tilt tricopter aerial vehicles. Tricopter UAVs have been shown to be more agile and manoeuvrable offering more advantages than other multicopters like the quadcopter for example. The dynamic models for both tricopters are derived from first principles and experimental data is used to obtain the actuator constants. In the case of the single-tilt tricopter, a control allocation algorithm is also proposed to solve the problem of the number of control inputs being more than the number of actuators since the single-tilt tricopter has only four actuators (3 rotors and 1 servo) and a greater number of control inputs (forces and torques). A cascaded-PID control scheme is then used to stabilize the single-tilt tricopter in hover mode. The simulation results yield realistic control inputs and the outputs have acceptable performance. The feasibility of the proposed scheme is then validated with some experiments on the developed tricopter platform in hover.

For the multi-tilt tricopter, a Quaternion Feedback Control (QFB) scheme is proposed which uses unit quaternions to represent the attitude dynamics in order to avoid gimbal lock which occurs when the pitch angle approaches  $\pm 90$  degrees if using Euler angles. Also, a linear Model Predictive Control (MPC) scheme which uses the Linear Parameter Varying model of the tricopter is proposed. These control techniques are tested and compared using simulations in Matlab/Simulink. The feasibility of achieving independent position and attitude control, with possibility of translating in the longitudinal and lateral directions without changing the multi-tilt tricopter's attitude, is shown in simulation and demonstrated in experiments on the in-house tricopter. The hardware implementation of this concept is achieved by developing an algorithm using the PX4 framework which allocates the lateral and longitudinal forces via mapped transmitter knobs.

Finally, this thesis also proposes a robust leader-following formation control scheme for a class of multi-agent systems that can be modelled as a group of networked closed-loop linearized multi-tilt tricopter agents utilizing Negative Imaginary (NI) theory. A continuous time subspace identification method for NI systems based on the Laguerre filter is proposed, which guarantees that the resultant model is NI. Sliding Mode Control (SMC) is used to linearize the tricopter system in closed-loop having six inputs and outputs instead of the more common feedback or Jacobi linearization methods. The usefulness of the identification algorithm is shown via the identification of an NI model for the SMC-linearized tricopter (inner-loop). The leader-following formation problem is formulated as an asymptotic tracking problem of a distributed Strictly Negative Imaginary (SNI) plus Very Strictly Passive (VSP) system being cascaded with a network of closed-loop linearized multi-tilt tricopter agents. An in-depth simulation case study is performed on a formation tracking mission for a group of six SMC-linearized multi-tilt tricopters and the results show that the tricopter agents achieve consensus tracking and leader-following group formation tracking.

# Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

# Copyright Statement

- i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.
- ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.
- iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.
- iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see <http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420>), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see <http://www.manchester.ac.uk/library/aboutus/regulations>) and in The University’s Policy on Presentation of Theses.

# Dedication

To my parents, Professor Abara E. Abara & Mrs. Veronica Abara...

# Acknowledgements

I would like to use this opportunity to thank everyone who has assisted me in one way or another during the course of my PhD studies.

I would like to express my sincere gratitude to my supervisor, Professor Alexander Lanzon for giving me the opportunity to work under his supervision, for always finding time to fit me in to his busy schedule and for his detailed and insightful contributions to my work. I also thank him for his guidance and support which helped me grow to be a better person and more independent researcher.

I would also like to appreciate my first and second year examiners, Dr. Long Zhang and Dr. Ognjen Marjanovic for their inputs which steered my work in the right direction.

I must thank Dr. Parijat Bhowmick for his immense support and contributions to my work. Many thanks also to Dr. Somasundar Kannan for all his advice and contributions especially with the experimental aspects of my research. I will always remain grateful to you both.

My sincere thanks to the present and past members of our research team: Dr. Gabriela Salcan-Reyes, Dr. Ola Skeik, Dr. Junyan Hu, Suleiman Kurawa, Peter Hilborne and Yu-Hsiang Su, for all their support and contributions and for all the fun times we shared together.

Special thanks to my family: my wife Monica, although she joined me at a later stage of my studies, she has been very supportive and understanding, I owe you my deepest gratitude; my parents, Professor & Mrs. Abara, my sister and brothers, Esther, Charles and Joseph for their unflinching support and prayers throughout my studies.

Finally, I am grateful to my sponsors, the Petroleum Technology Development Fund (PTDF) of Nigeria, for affording me the opportunity to pursue my Doctorate.

# Publications

The contents of Chapters 3, 4 and 5 have been published or submitted for publication respectively, in the following:

- D. Abara, S. Kannan, and A. Lanzon, “Development and stabilization of a single-tilt tricopter with optical flow for GPS-denied environments,” in *Proceedings of the 21st IFAC World Congress*, vol. 53, Berlin, Germany, pp. 8897–8902, July 2020.
- D. Abara, P. Bhowmick, and A. Lanzon, “Development and control of Multirotor-tilting tricopter with direct actuation for position control”, *under review in International Journal of Control*, December, 2021.
- D. Abara. P. Bhowmick, and A. Lanzon, “A negative imaginary theory based robust formation control scheme for a group of networked multi-tilt tricopters applying sliding-mode inner-loop control”, *under review in Automatica*, February, 2022.

# Nomenclature

## Field of numbers

$\mathbb{R}$	real numbers
$\mathbb{R}^n$	real column vectors with $n$ entries
$\mathbb{R}^{m \times n}$	real matrices with $m$ rows and $n$ columns
$\mathbb{C}$	complex numbers
$\mathbb{C}^n$	complex column vectors with $n$ entries
$\mathbb{C}^{m \times n}$	complex matrices with $m$ rows and $n$ columns
$\mathbb{F}^{m \times n}$	real or complex matrices with $m$ rows and $n$ columns
$\mathbb{R}_{>0}$	positive real numbers
$\mathbb{R}_{\geq 0}$	non-negative real numbers

## Relational symbols

$<$	less than
$>$	greater than
$\forall$	for all
$\geq$	greater than or equal
$\in$	belong to
$\leq$	less than or equal
$\Leftrightarrow$	is equivalent to
$\neq$	not equal to
$\nRightarrow$	not equivalent to
$\Rightarrow$	implies
$\rightarrow$	tends to
$\subset$	is a proper subset of



$\triangleq$  is defined as

### Matrix operations

$\mathbf{I}_n$	$n \times n$ identity matrix
$A^\top$	transpose of $A$
$A^*$	complex conjugate transpose of $A$
$\bar{A}$	complex conjugate of $A$
$A^{-1}$	inverse of matrix $A$
$A^{-\top}$	shorthand for $(A^{-1})^\top$
$A^{-*}$	shorthand for $(A^{-1})^*$
$\hat{\mathbf{u}}_i$	$i$ -th unit vector $[1 \ 0 \ 0]^\top$
$\hat{\mathbf{u}}_j$	$j$ -th unit vector $[0 \ 1 \ 0]^\top$
$\hat{\mathbf{u}}_k$	$k$ -th unit vector $[0 \ 0 \ 1]^\top$
$A > 0 (\geq 0)$	$A$ is positive definite (semidefinite)
$a_{ij}$	the $i$ -th row and $j$ -th column of a real matrix
$\bar{a}_{ij}$	the $i$ -th row and $j$ -th column of a complex matrix
$\det(A)$	determinant of $A$
$\lambda_{\max}(A)$	maximum eigenvalue of matrix $A$
$\text{rank}(A)$	rank of matrix $A$

### Miscellaneous

$\mathbf{v}^b$	vector $\mathbf{v}$ given relative to the body-fixed frame
$\mathbf{v}^e$	vector $\mathbf{v}$ given relative to the earth (inertial) frame
$\mathbf{v}^{ln}$	vector $\mathbf{v}$ given relative to the local frame $n$
$\circ$	quaternion product
$\otimes$	kronecker product
$c_\phi$	shorthand for $\cos \phi$
$s_\phi$	shorthand for $\sin \phi$
$t_\phi$	shorthand for $\tan \phi$
$\mathbf{R}_e^b$	rotation matrix which transforms a vector from earth to body-fixed coordinates
$\mathcal{RH}_\infty^{m \times n}$	set of all proper, real, rational and asymptotically stable transfer function

	matrices of dimensions $(m \times n)$
$\mathcal{SO}(3)$	special orthogonal group of all $(3 \times 3)$ rotation matrices
$\text{diag}(a, b, c)$	diagonal matrix with diagonal entries $a, b, c$
■	end of proof

## Acronyms

BLDC	Brushless Direct Current
BR	Bounded Real
CAD	Computer Aided Design
CAN	Controller Area Network
CT	Continuous Time
DC	Direct Current
DOF	Degree of Freedom
DT	Discrete Time
ESC	Electronic Speed Controller
GPS	Global Positioning System
I2C	Inter-Integrated Circuit
LiPo	Lithium Polymer
LMI	Linear Matrix Inequality
LPV	Linear Parameter Varying
LTi	Linear Time Invariant
MAS	Multi-Agent Systems
MIMO	Multi Input Multi Output
MPC	Model Predictive Control
NI	Negative Imaginary
PDB	Power Distribution Board
PID	Proportional Integral Derivative
PR	Positive Real
PWM	Pulse Width Modulation
QFB	Quaternion Feedback
RAM	Random Access Memory
REF	Row Echelon Form

RTOS	Real-time operating system
SISO	Single Input Single Output
SMC	Sliding Mode Control
SNI	Strictly Negative Imaginary
UART	Universal Asynchronous Receiver/Transmitter
UAV	Unmanned Aerial Vehicle
VSP	Very Strictly Passive
VTOL	Vertical Take-off and Landing

# Chapter 1

## Introduction

### 1.1 Background and Motivation

Unmanned Aerial Vehicles (UAVs) may be classed as either fixed wing or rotary wing [1]. The fixed wing UAVs require a runway for take-off while the rotary wing UAVs have the advantage of vertical take-off and landing (VTOL) and this has led to an increased interest in VTOL aircraft which are sometimes called multi-rotors or multicopters typically named based on the number of rotors on the aircraft. This interest in multicopters is due to their ability to operate in hazardous or difficult terrains without needing a runway as a result of the VTOL feature. Inspired by these reasons, the focus of this thesis is on control of multicopters.

Multicopters may exist in various configurations and may be classed as under-actuated, fully-actuated or over-actuated. An important concept in the description of dynamical systems which is essential in understanding these different classifications is the degrees-of-freedom (DOF). The number of DOF is equal to the number of coordinates which are used to specify the configuration of a dynamic system minus the number of independent equations of constraint [2, 3]. In other words, the DOF is the number of independent generalized coordinates that completely defines the configuration of a dynamic system at any time instant. The typical multicopter has 6-DOF [4] which are the three translational positions (lateral, longitudinal and altitude) and three attitude angles (roll, pitch and yaw) which together describe the position and orientation of the multicopter in three-dimensional space. The physical configuration of the multicopter affects how many of these DOF can be directly actuated or equivalently,

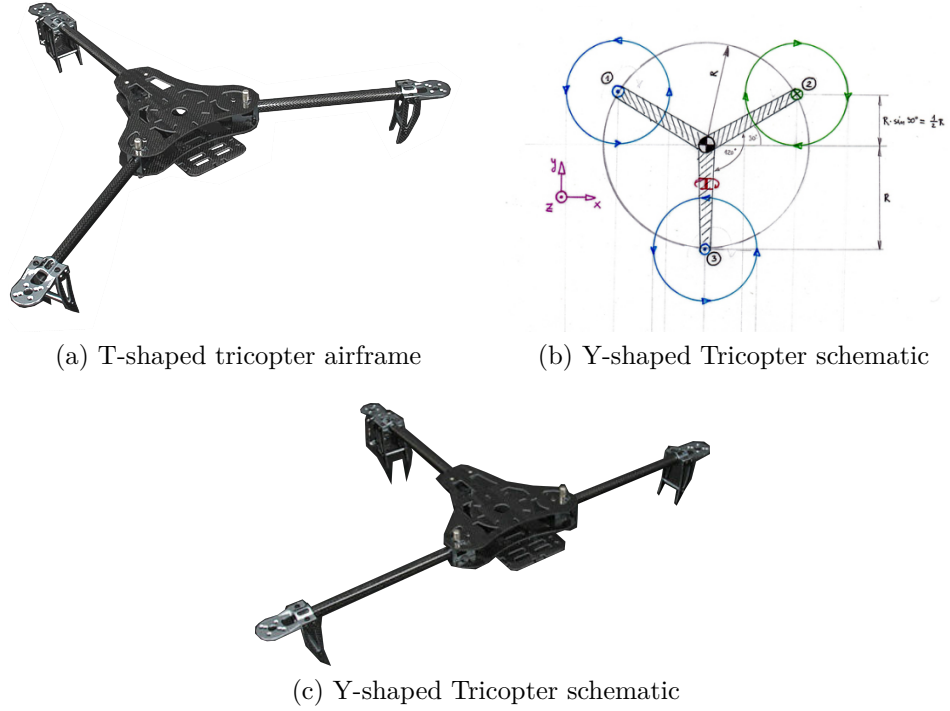


Figure 1.1: Tricopter Configurations [images from Google].

how many control inputs (forces and torques) are available to directly control the multicopter's DOF. It is therefore necessary to understand how the available control inputs relate to the number and arrangement of the actuators (motors and servos) of the multicopter system and this relationship is often called a control allocation. The system is considered under-actuated if the number of available control inputs is less than the number of DOF [3, 5]. This implies that the number and arrangement of the multicopter's actuators results in a configuration that is not sufficient to fully actuate all the 6-DOF. Most classic multicopters are under-actuated. For example, the classic quadcopter [6] is under-actuated because only 4 out of the 6-DOF namely the altitude, roll, pitch and yaw, can be directly and independently controlled. Control of the lateral and longitudinal positions is achieved indirectly by controlling the attitude. A higher number of actuators does not necessarily indicate higher DOF. For example, the classic hexacopter [7] has six rotors but is still under-actuated just like the quadcopter. If the number of available control inputs is equal to the number of DOF, such multicopter is said to be fully-actuated [8]. This implies that the arrangement of actuators is such that all 6-DOF can be independently controlled. Over-actuated multicopters have the same properties as fully-actuated ones but typically have more than 6 actuators thereby

providing more flexibility in the control allocation due to the redundant actuators [9]. This is often useful for fault-tolerance and energy efficiency.

Typically, the rotors of multicopters are in pairs so that inherent reaction torques are accounted for using counter-rotating propellers. In order to achieve more efficiency in terms of size, manoeuvrability and even costs, researchers have proposed several configurations of multicopters. A popular configuration is the quadcopter which has four rotors and the attitude is controlled by varying the speeds of the opposite rotors while lift is generated by the thrust produced by the four rotors rotating at equal speed [6]. However, most quadcopters have the downside of being rigid due to their airframe structure. That is, it is difficult to achieve and sustain large attitude angle changes during flight making quadcopters less flexible or agile. It is possible to perform instant manoeuvres such as flips for example, but such motions cannot be sustained for continuous periods. The exception to this is where the quadcopter has some or all of its rotors tilted. In this situation, the quadcopter becomes a tilt-rotor and is hence more flexible. This thesis therefore focuses on another less common configuration known as the tricopter which has 3 rotors, with one of the rotors (normally the tail rotor) allowed to tilt for direct yaw control. Although, the odd number of rotors leads to yaw imbalance making the tricopter unstable and control more difficult, the design is cheaper [10] than an equivalently sized quadcopter due to less power requirements from a reduction in number of motors, and the inherent instability and tilt mechanism makes the tricopter more agile and manoeuvrable, compared to the quadcopter [11]. In this thesis, the configuration with one rotor tilted is termed single-tilt (or classic) tricopter. In addition, the tricopter being a tilt-rotor airframe opens up opportunities for research into the tilt-rotor space. These are the reasons this thesis focuses on the tricopter rather than the more common quadcopter.

A T-shaped configuration modelled using Newton-Euler methods was proposed by [12] and a nonlinear control technique which utilized nested saturations was used to achieve stability of the single-tilt tricopter. A similar but different T-shaped single-tilt tricopter was proposed in [13] where the UAV has the combined features of both fixed-wing and rotary-wing UAVs. The three motors point upwards like any standard multicopter on take-off, thereby providing the needed thrust to gain altitude like a rotary-wing (VTOL) UAV. After take-off, the two front rotors are tilted forwards,

to vector the thrust from these rotors in the longitudinal direction. As a result, this single-tilt tricopter setup achieves forward motion like a fixed-wing UAV. This is similar to the Bell Boeing V-22 Osprey convertiplane. A more complex airframe is proposed by [11] where instead of using a tilted tail rotor, the yaw angle is controlled by a pair of flaps mounted on the slipstream of the propellers. While the idea is innovative, the complexity makes it less intuitive. Noting that the tail servo is the weakest link in the single-tilt tricopter airframe, a fault-tolerant controller is proposed in [14] to combat the stuck fault of the tail servo. Speaking of control techniques, a number of linear and nonlinear control techniques have been applied to solve the control problem of the single-tilt tricopter. Besides the use of nested saturations in [12], fuzzy-logic has been used in [15] where a similar model to [13] is proposed. An MPC-based control technique is applied in [16] to achieve stability of the single-tilt tricopter's position. In addition, a control algorithm is proposed therein. Similar to [16], a nonlinear MPC scheme is used in [17] to control the single-tilt tricopter and online learning capabilities of the proposed method are investigated with simulations. The work of [18] makes use of adaptive fuzzy gain scheduling to tune PID controllers for the single-tilt tricopter, thus yielding better performance than conventional PIDs. It can be observed that there are still opportunities for further research into the single-tilt tricopter especially in the area of experimental validation of the tricopter concept with physical hardware as most of the existing works are based mainly on simulations. Also, in the current literature for the single-tilt tricopter, models derived from first principles are mostly used rather than models identified from experimental data. Consequently, the practicability of the existing methods with regards to real hardware has received little consideration. These issues open up opportunities for additional research into the tricopter's airframe structure or design and into the development of control techniques for the tricopter.

Motivated by these concerns together with a desire to gain further insight into the tricopter problem in general, the hardware for a single-tilt tricopter using open-source tools based on the PX4 [19] framework is constructed in this thesis. The model is derived from first principles and to make the control as practicable as possible, varied experiments are developed and used in identifying the model parameters. As the single-tilt tricopter is under-actuated, a control allocation scheme which allocates the actuator signals is proposed to solve this problem. Cascaded-PID control is used to

study the practicability of the scheme with both simulations and trial experiments.

Another tricopter configuration which is investigated in this thesis is the multi-tilt tricopter. In this configuration, all three rotors are tilted thereby providing more benefits such as further increasing the agility of the tricopter and offering the possibility of achieving independent translational and rotational motions. That is, making it possible to control the position (with respect to the inertial frame) of the tricopter without changing its attitude. This configuration is advantageous for example for mounting a camera directly on the UAV body rather than on a gimbal. Since this configuration offers greater tilting capabilities, a variety of views maybe obtained by directly varying the attitude of the UAV, thereby varying the views of the camera. In such situation, the use of a gimbal for changing the camera's view becomes unnecessary. The multi-tilt tricopter has gained less attention compared to the single-tilt tricopter. One of the pioneering works on the multi-tilt tricopter is the work of [20] where a novel airframe in which all rotors can tilt was proposed. The authors used feedback linearization and  $\mathcal{H}_\infty$ -control to show how the attitude could be stabilized. A similar tricopter where all rotors can tilt is studied in [21] and a so-called pilot-supporting controller is proposed to control the vehicle, and more recently in [22], flatness-based control has been applied to show how a tricopter with all rotors independently tilting can follow arbitrary trajectories in air. Another interesting work relating to the multi-tilt tricopter is that of [23] where all rotors independently tilt. However, apart from the fact that the tail rotor points downwards in the reverse direction, the airframe is T-shaped rather than Y-shaped as compared to [20–22] and PID control is used to verify the concept with flight experiments. Besides the fact that there has been fewer studies on the multi-tilt tricopter problem, a similar observation to the single-tilt case can be made relating to the fact that there has been more focus on models based on first principles rather than experimental models, and the models used in practically all the former works mentioned represent rotational dynamics (attitude) using Euler angles which are prone to gimbal lock issues [24]. Gimbal lock is an effect which occurs when the pitch angle reaches  $\pm 90^\circ$  leading to singularities in the computations and subsequent processing errors especially when implementing on hardware. This is an important factor to consider for the multi-tilt tricopter since it is a very agile and flexible UAV where the changes in attitude angles may be large, as a result of



the airframe configuration. In addition, none of the existing works have explored a quaternion feedback control technique for the tricopter problem in general.

Motivated by these issues and the opportunities which the multi-tilt airframe configuration presents, a tricopter with independently tilting rotors is also constructed in this thesis using open source tools (PX4) similar to the single-tilt case resulting in a fully actuated vehicle. A custom tilt-mechanism is developed and used to tilt each of the rotors in a wider range for more flexibility. Different from existing literature on tricopters, this thesis proposes a quaternion-feedback control technique for attitude stabilization. The model used in this case makes use of quaternions in representing the attitude thereby avoiding gimbal lock. Like the single-tilt case, experiments are used to obtain the parameters of the model. Besides the quaternion-feedback scheme, a linear MPC scheme is also proposed for attitude stabilization of the multi-tilt tricopter. Here, rather than linearizing the model about operating points as is standard practice, a Linear-Parameter-Varying (LPV) model is used instead. This offers better precision in terms of the validity of the model since the nonlinearities of the tricopter are captured in the LPV model. As a result, by using this LPV model, even though the multi-tilt tricopter system is inherently nonlinear, a linear control method can be applied to control it. This thesis also shows via simulations, the capability of the multi-tilt tricopter to perform independent translational motion in the longitudinal and lateral directions without changing attitude. This concept is then demonstrated by implementing the control forces for the longitudinal and lateral positions in the control allocation of the PX4 flight stack mixing system, validated using bench test experiments on the multi-tilt tricopter test bed. Note that a radio transmitter typically supports the control of only roll, pitch, yaw and thrust. This control allocation implemented in PX4 is necessary in order to additionally control the longitudinal and lateral positions of the tricopter using the radio transmitter.

### 1.1.1 Multi-agent Systems

The cooperative control of multi-agent systems (MAS) has also drawn a decent amount of attention in recent years. In this context, a number of systems or (UAVs in this case) are considered as agents forming a network where information can be exchanged between agents via the network [25]. Although each individual agent may have limited

resources, the network of agents as a whole can perform complex tasks in a coordinated manner. Researchers from multiple disciplines have shown interest in MAS due to their wide applications in multirobot cooperation [4, 26, 27] and distributed sensor networks [28, 29] to name a few. As noted in [30], the MAS domain includes three main areas namely consensus control [31, 32], containment control [33] and formation control [34]. Of these three, formation control has witnessed an immense growth over the past decade [35]. In many cases, if formation control is achieved, consensus control is inherently achieved and this may account for the growth and amount of research in the formation control space. The formation control problem is to develop control techniques or strategies to achieve group formation tracking. That is, to make the agents achieve a specified formation (or shape) with reference to a target.

Recognising that MAS play a key role in today's world with substantial amount of research in this space, this thesis also investigates the formation control for a group of multi-tilt tricopters. The literature on MAS suggests that formation control has seen the highest amount of interest, of the domains in the MAS field [35]. This has been achieved using several techniques and approaches [30, 36–40] including an approach called Negative Imaginary (NI) systems theory [41, 42]. NI systems theory is a robust control method for achieving robust stability of a feedback interconnection for a certain class of systems. The Nyquist stability criterion [43] provides stability conditions under which a feedback interconnection is robustly stable using a lot of information about the interconnected systems. If less information is known about the interconnected systems, it is still possible to guarantee robust stability if certain conditions are known. For instance, if the interconnected systems are Bounded Real (BR) or Positive Real (PR), then concepts such as small-gain theorem [44] or passivity theorem [45] respectively, provide conditions for establishing the robust stability of the feedback interconnection. The NI theory provides a result of similar nature where a certain class of systems may be characterized as having NI frequency response. Consequently, the NI theory provides necessary and sufficient conditions for establishing the stability of an interconnection of such systems [46, 47], using only limited information about the interconnected systems. The NI theory itself has gained a lot of traction since inception and has been shown to have many useful engineering applications including Nanopositioning, vehicle platooning, vibration control, control of large space structures, flexible robotic arms,

DC machines and multi-agent networked systems (see [48–53] and references therein). In general, NI theory is appealing due to its simple internal stability condition<sup>1</sup> that depends on the loop gain at zero frequency only. This thesis focuses mainly on the application of the NI theory to multi-agent systems.

In [55], a consensus-based formation control framework is designed for a multi-vehicle system using NI theory. Obstacle avoidance and detection is also investigated therein and compared to the similar work of [56] which is based on artificial potential field function control. The authors in [56] and [57] extended the works of [31] and [32] to develop a formation control framework for a group of heterogeneous vehicles with time-invariant switching. Two-wheeled mobile robots interacting through a directed and balanced graph are made to achieve a rendezvous control scheme in [58], also using NI theory. The idea of using the NI/SNI stability result to develop cooperative control schemes is due to the fact that a certain class of systems (such as UAVs, mobile robots for example) can be modelled as a group of networked single or double integrator agents which satisfy the NI property. This single or double integrator systems are typically achieved by feedback linearization of the nonlinear dynamics of the original systems. Another factor is that a number of studies in literature [41, 55, 57], have shown specifically that linearized dynamics of the inner-loop of a class of UAVs in particular, exhibit the NI property. Thus, a distributed SNI controller may be used to robustly stabilize such UAVs.

Motivated by these existing results and in an effort to further develop the formation control of multi-agent UAVs, this thesis proposes a leader-following formation control scheme for a network of multi-tilt tricopter UAVs. Different from existing works such as [20] where feedback linearization has been used to obtain single/double integrator dynamics, this thesis proposes Sliding Mode Control (SMC) for linearizing the nonlinear dynamics of the multi-tilt tricopter. The result is a SMC-linearized closed-loop system with six inputs (three positions with respect to the inertial frame and three attitude angles) and six corresponding outputs. Noting from [57] that linearized UAV dynamics in closed-loop can be modelled as NI systems and in an attempt to obtain the dynamics of the SMC-linearized multi-tilt tricopter model, this thesis develops a closed-loop

---

<sup>1</sup>A necessary and sufficient condition for the internal stability of a positive feedback interconnection of NI and Strictly NI systems, say  $M(s)$  and  $N(s)$ , is  $\lambda_{\max}[N(0)M(0)] < 1$  [47, 54].

system identification algorithm that guarantees that the identified model is NI. The algorithm is based on the classic subspace identification method but makes use of the Laguerre filter [59] and extends the work of [60]. The developed scheme is used to obtain transfer function models for each channel of the SMC-linearized multi-tilt tricopter and frequency domain analysis is used to confirm that these models are in fact NI. These models when combined form a six-by-six transfer function matrix describing the full SMC-linearized dynamics of each of the multi-tilt tricopters in a network, and which serves as the inner loop. Consequently, an outer-loop output feedback distributed controller is proposed to achieve formation control. This distributed controller is based on the NI and passivity theories via a derived closed-loop stability result which establishes that a network of SMC-linearized multi-tilt tricopters being NI or SNI can be robustly stabilized with negative feedback. To prove the stability of the states, the characteristics of the eigenvalue loci of networked NI and SNI systems is used instead of the Lyapunov-based approach which is more common in literature. Simulation studies are used to demonstrate the effectiveness and usefulness of the proposed methods.

## 1.2 Contributions to the state-of-the-art

The contributions of this thesis to the state-of-the-art are as follows:

- Design and construction of servo tilt-mechanism which extends the tilt angle range of the rotors in a multi-tilt tricopter.
- The development of control allocation scheme which allocates the actuator signals via a non-square mixer matrix due to under-actuation in the single-tilt tricopter.
- Demonstration of capability of multi-tilt tricopters to translate in longitudinal and lateral directions without change in attitude.
- Development of direct (manual) actuation method to control lateral and longitudinal motion of the multi-tilting tricopter via thrust vectoring, using a transmitter.
- Development of a novel system identification method for NI systems in continuous time.

- Development of closed-loop stability result for a network of NI/SNI systems in feedback with mixed NI+VSP controllers.
- Design of distributed control law using proposed stability result to achieve cooperative control with both consensus and formation control.

## 1.3 Thesis Organization

The rest of this thesis is organized as follows:

### **Chapter 2 - Preliminaries:**

This chapter presents some relevant technical background which is essential for the reader to understand the rest of the thesis. The concepts introduced include Linear algebra, Linear systems theory, Graph theory and Negative imaginary systems theory. A review of coordinate systems, Quaternions, Euler angles and rotation matrices is also provided.

### **Chapter 3 - Development and stabilization of a low-cost single-tilt tricopter:**

In this chapter, a low-cost single-tilting tricopter aerial vehicle is developed with optical flow estimation for indoor navigation. A dynamic model is derived and experimental data is used to obtain the actuator constants. A CAD model is then developed and is used to obtain the moments of inertia with respect to the three main axes. A control allocation algorithm is also proposed to solve the problem of the number of control inputs being more than the number of actuators since the single rotor tilt tricopter has only four actuators (3 rotors and 1 servo). A cascaded-PID control scheme is then used to stabilize the tricopter in hover mode. The simulation results yield realistic control inputs and the outputs have acceptable performance. The feasibility of the proposed scheme is then validated with some experiments on the developed tricopter platform in hover.

### **Chapter 4 - Development and control of Multirotor-tilting tricopter with direct actuation for position control:**

This chapter presents the development and control of a custom multirotor-tilting tricopter UAV. The dynamic model is derived and data from varied experiments

is used to identify the model parameters. A Quaternion Feedback Control (QFB) scheme is proposed which uses unit quaternions to represent the attitude dynamics in order to avoid gimbal lock which occurs when the pitch angle approaches  $\pm 90$  degrees if using Euler angles. Next, a Model Predictive Control (MPC) scheme which uses the Linear Parameter Varying model of the tricopter is proposed. Both control techniques are tested and compared using simulations. The feasibility of full actuation is shown through position control achieved by feedback linearization of the nonlinear translational equations to yield the desired control forces for the longitudinal and lateral directions. This idea of independently controlling the lateral and longitudinal positions is then extended to hardware through the development of a control allocation via thrust vectoring based on the PX4 framework. Real-time indoor experiments on the in-house tricopter test-bed are used to validate the control allocation for motion in the lateral and longitudinal directions.

**Chapter 5 - A negative imaginary robust formation control scheme for a group of networked tricopters over inner-loop sliding-mode control:**

This chapter proposes a robust leader-following formation control scheme for a class of multi-agent systems that can be modelled as a group of networked closed-loop linearized multi-tilt tricopter agents utilizing Negative Imaginary (NI) theory and Passivity Theory. A continuous time subspace identification method for NI systems based on the Laguerre filter is proposed, which guarantees that the resultant model is NI. The model of the tricopter is derived and Sliding Mode Control (SMC) technique is used to linearize the system in closed-loop having six inputs and outputs. The developed identification algorithm is used to identify an NI model for the SMC-linearized tricopter (inner-loop). According to NI theory, since the linearized tricopter is NI, if SNI controllers are used to control them, then robust stability for the whole network is achieved. The leader-following formation problem has been formulated as an asymptotic tracking problem of a distributed SNI+VSP system being cascaded with a network of closed-loop linearized multi-tilt tricopter agents. Eigenvalue loci technique is used instead of Lyapunov-based approach, to prove the asymptotic convergence of the formation tracking error. A simulation study for a group of six SMC-linearized multi-tilt tricopters is provided to show the usefulness and effectiveness of the proposed scheme.

**Chapter 6 - Concluding Remarks:**

This chapter concludes this thesis by providing a summary of the contributions herein, and highlighting possible areas for future research.

# Chapter 2

## Preliminaries

This chapter provides a brief but necessary technical background to prepare the reader for the chapters that follow. This chapter is intended to be used as a quick reference for understanding the derivations of the results in subsequent chapters.

### 2.1 Linear Algebra

This section contains some basic linear algebra definitions and properties which are used in this thesis.

#### 2.1.1 Linear subspaces

The range and rank of a matrix are important properties used in linear algebra and are respectively defined as follows:

**Definition 2.1** (Range, [44, 61]). Let  $A \in \mathbb{F}^{m \times n}$ . Then, the range or image of  $A$  denoted  $\text{Im}(A)$  is defined as

$$\text{Im}(A) = \{y \in \mathbb{F}^m : y = Ax, x \in \mathbb{F}^n\}.$$

**Definition 2.2** (Rank, [44, 61]). Let  $A \in \mathbb{F}^{m \times n}$ . Then the rank of  $A$  is defined as

$$\text{rank}(A) = \dim(\text{Im}(A)).$$

Since  $\text{rank}(A) = \text{rank}(A^*)$ , then  $\text{rank}(A)$  equals the maximum number of *linearly independent* rows and columns. One way of obtaining the rank of a matrix is by row



reduction (Gaussian elimination) to reduce the matrix to its Row Echelon Form (REF), see [62] for details. The following definition applies the rank condition to a special case of when the matrix  $A$  is square.

**Definition 2.3** ([62]). Given a square matrix  $A \in \mathbb{F}^{n \times n}$ . If  $\text{rank}(A) = n$ , then  $A$  is said to be *full rank* and is nonsingular. That is, its inverse  $A^{-1}$  exists.

This will be used later to show that a matrix is always invertible as far as it is full rank. The notion of Unitary matrices is another concept which applies to several applications including coordinate rotations which are considered in this thesis. Thus, a formal definition is given as follows.

**Definition 2.4** (Unitary matrix, [44]). A square matrix  $U \in \mathbb{F}^{n \times n}$  whose columns form an orthonormal basis for  $\mathbb{F}^n$  is called a unitary matrix (or orthogonal matrix if  $\mathbb{F} = \mathbb{R}$ ), and it satisfies  $U^*U = I = UU^*$ .

### 2.1.2 Symmetric and semidefinite matrices

The concept of semidefinite matrices has been used in many applications and is essential in the development of NI theory which forms a part of this thesis. A formal definition is given as follows.

**Definition 2.5** (Symmetric matrix, [62]). Let  $A \in \mathbb{R}^{n \times n}$  (respectively  $\in \mathbb{C}^{n \times n}$ ), then  $A$  is said to be symmetric (respectively hermitian) if  $A = A^\top$  (respectively  $A = A^*$ ) that is,  $a_{ij} = a_{ji}$  (respectively  $a_{ij} = \bar{a}_{ji}$ ).

**Definition 2.6** (Skew-symmetric matrix, [62]). Let  $A \in \mathbb{R}^{n \times n}$  (respectively  $A \in \mathbb{C}^{n \times n}$ ), then  $A$  is said to be skew-symmetric (respectively skew-hermitian) if  $A = -A^\top$  (respectively  $A = -A^*$ ) that is,  $a_{ij} = -a_{ji}$  (respectively  $= -\bar{a}_{ji}$ ).

**Definition 2.7** (Semidefinite matrix, [44]). A square Hermitian matrix  $X = X^*$  is said to be positive (semi) definite, denoted by  $X > 0$  ( $X \geq 0$ ), if  $x^*Xx > 0$  ( $x^*Xx \geq 0$ )  $\forall x \neq 0$ .

### 2.1.3 Cross products

The cross product is an important property with applications in many areas including coordinate rotations which are used in this thesis when modelling the tricopter. A

formal definition is as follows.

**Definition 2.8.** [63] The *cross-product* of  $\mathbf{u} \in \mathbb{R}^3$  and  $\mathbf{v} \in \mathbb{R}^3$ , denoted by  $\mathbf{u} \times \mathbf{v}$ , is a vector  $\mathbf{w} \in \mathbb{R}^3$  that is normal to the plane of  $\mathbf{u}$  and  $\mathbf{v}$  and is in a direction such that  $\mathbf{u}, \mathbf{v}, \mathbf{w}$  (in that order) form a right-handed system. The length of  $\mathbf{w}$  is defined to be  $|\mathbf{u} \times \mathbf{v}| = |\mathbf{u}||\mathbf{v}| \sin \theta$ , where  $\theta$  is the angle between  $\mathbf{u}$  and  $\mathbf{v}$ .

It is convenient to represent the cross-product as a matrix function given by,

$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \mathbf{S}(\mathbf{u})\mathbf{v}, \quad (2.1)$$

where  $\mathbf{S}(\cdot) \in \mathbb{R}^{3 \times 3}$  is a skew-symmetric *cross product* matrix function. An important property of the cross product is that it is anti-commutative, that is,

$$\begin{aligned} \mathbf{u} \times \mathbf{v} &= -(\mathbf{v} \times \mathbf{u}), \\ \mathbf{S}(\mathbf{u})\mathbf{v} &= -\mathbf{S}(\mathbf{v})\mathbf{u}. \end{aligned} \quad (2.2)$$

### 2.1.4 Kronecker products

The Kronecker product is an important concept that has been used in this thesis to express a networked multi-agent system. Its definition and key properties are given as follows.

**Definition 2.9** (Kronecker product, [44]). The *Kronecker product* of  $A \in \mathbb{F}^{m \times n}$  and  $B \in \mathbb{F}^{p \times q}$  is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix} \in \mathbb{F}^{mp \times nq}, \quad (2.3)$$

and has the following properties:

$$A \otimes (B + C) = A \otimes B + A \otimes C, \quad (2.4)$$

$$(kA) \otimes B = A \otimes (kB) = k(A \otimes B) \text{ where } k \text{ is a scalar}, \quad (2.5)$$

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD), \quad (2.6)$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}. \quad (2.7)$$

### 2.1.5 Eigenvalues and Eigenvectors

The eigenvalues of a square matrix provide important information about the matrix, and therefore play a key role in analysis and design of linear control systems which is done in this thesis. A brief definition of eigenvalues and eigenvectors is as follows.

**Definition 2.10** (Eigenvalues, [61]). Let  $A \in \mathbb{C}^{n \times n}$ . If a scalar  $\lambda$  and a non-zero vector  $x$  satisfy the equation

$$Ax = \lambda x, \quad x \in \mathbb{C}^n, \quad x \neq 0, \quad \lambda \in \mathbb{C},$$

then  $\lambda$  is called an *eigenvalue* of  $A$  and  $x$  is called an *eigenvector* of  $A$  associated with  $\lambda$ . The pair  $\lambda, x$  is an *eigenpair* for  $A$ .

## 2.2 Linear Systems Theory

This section presents some concepts in linear systems theory and their properties with a focus on linear control systems. The material in this section can be found in [44].

### 2.2.1 Description of systems

Consider the following linear time-invariant (LTI) system,

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (2.8)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$ ,  $D \in \mathbb{R}^{q \times m}$ , and  $x(t) \in \mathbb{R}^n$ ,  $u(t) \in \mathbb{R}^m$  and  $y(t) \in \mathbb{R}^p$  represent the state, input and output respectively. If  $m = p = 1$ , the system (2.8) is said to be a SISO system while if  $m, p > 1$ , the system (2.8) is said to be a MIMO system. The corresponding transfer function (transfer matrix for MIMO case) is defined as  $M(s) = C[sI - A]^{-1}B + D$ . The eigenvalues of  $A$  in system (2.8) are the poles of  $M(s)$ .

The system (2.8) is often represented in compact form by the notation

$$M(s) = \left[ \begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] = C[sI - A]^{-1}B + D \quad (2.9)$$

where  $M(s)$  is a transfer function (or matrix). An important operation which can be performed on  $M(s)$  and which is used extensively in NI theory is the transpose given below.

**Definition 2.11.** The transpose of the transfer matrix (2.9) is given by

$$M^\top(s) = B^\top(sI - A)^{-\top}C^\top + D^\top = \left[ \begin{array}{c|c} A^\top & C^\top \\ \hline B^\top & D^\top \end{array} \right]. \quad (2.10)$$

### 2.2.2 Stability, Controllability and Observability

The stability is an important concept in the analysis of control systems and the following definition provides a condition for the stability of system (2.8).

**Definition 2.12.** A matrix  $A$  is said to be *asymptotically stable* or *Hurwitz* if all its eigenvalues are in the open left half plane that is, all its eigenvalues have strictly negative real part.

The controllability and observability are very important in the design of feedback controllers for linear systems and they are applied in the design of Model Predictive and output feedback controllers in this thesis. The following lemmas provide useful results for checking these conditions.

**Lemma 2.1.** If  $\text{rank}[B, AB, \dots, A^{n-1}B] = n$ , then  $(A, B)$  is controllable.

**Lemma 2.2.** If  $\text{rank}[C^\top, A^\top C^\top, \dots, (A^{n-1})^\top C^\top] = n$ , then  $(C, A)$  is observable.

There are cases where a system does not satisfy the controllability or observability conditions but it may satisfy a reduced condition of being stabilizable or detectable respectively.

**Definition 2.13.** If there exists a matrix  $K \in \mathbb{R}^{m \times n}$  such that  $A + BK$  is Hurwitz, then system (2.8) is stabilizable or  $(A, B)$  is stabilizable.

**Definition 2.14.** If there exists a matrix  $L \in \mathbb{F}^{n \times p}$  such that  $A + LC$  is Hurwitz, then system (2.8) is detectable or  $(A, C)$  is detectable.

### 2.2.3 State space realizations

Most of the NI lemmas have been derived under the assumption of a minimal state-space realization. If the  $A$  matrix of  $M(s)$  in (2.9) has the least possible dimension, then the system is minimal and is given by the following result.

**Theorem 2.1.** *A state space realization  $\left[ \begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]$  of  $M(s)$  is minimal if and only if  $(A, B)$  is controllable and  $(C, A)$  is observable.*

### 2.2.4 Internal stability

Given a positive feedback interconnection as in Figure 2.1.

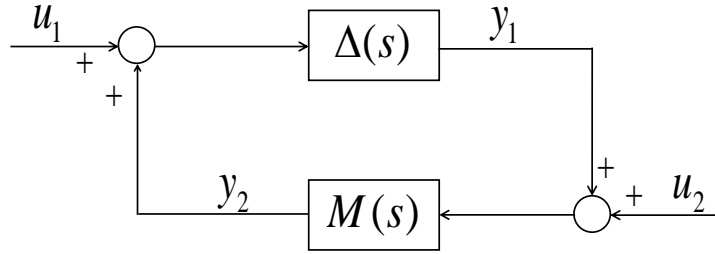


Figure 2.1: Positive Feedback interconnection of two systems  $\Delta(s)$  and  $M(s)$

**Lemma 2.3.** [44] *The feedback interconnection in Figure 2.1 is well-posed if and only if  $I - \Delta(\infty)M(\infty)$  is invertible.*

If for all bounded inputs  $(u_1, u_2)$ , the outputs  $(y_1, y_2)$  are also bounded, then the positive feedback interconnection in Figure 2.1 is internally stable. The following lemma provides a matrix characterization for internal stability. For convenience,  $\Delta(s)$  and  $M(s)$  will be denoted as  $\Delta$  and  $M$  respectively.

**Lemma 2.4.** [44] *The system in Figure 2.1 is internally stable if and only if the transfer matrix*

$$\begin{bmatrix} I & -M \\ -\Delta & I \end{bmatrix}^{-1} = \begin{bmatrix} I + M(I - \Delta M)^{-1}\Delta & M(I - \Delta M)^{-1} \\ (I - \Delta M)^{-1}\Delta & (I - \Delta M)^{-1} \end{bmatrix} \quad (2.11)$$

from  $(u_1, u_2)$  to  $(y_1, y_2)$  is  $\mathcal{RH}_\infty$ .

## 2.3 Graph theory

Graph theory is fundamental in the study of cooperative and consensus control of multi-agent systems which are considered in this thesis. This section therefore presents some background on graph theory as applied in this thesis and the material in this section can be found in [64–67].

Graphs are used to model information exchange among the agents in a network and the nodes in a graph represent the agents while the edges represent the information exchange among the agents. Graphs may be undirected where the flow of information is bidirectional (each agent can both send and receive information) or directed where the flow of information is directional (some agents can only receive information while others can only send information) as given in Figure 2.2. This thesis uses the directed graph topology.

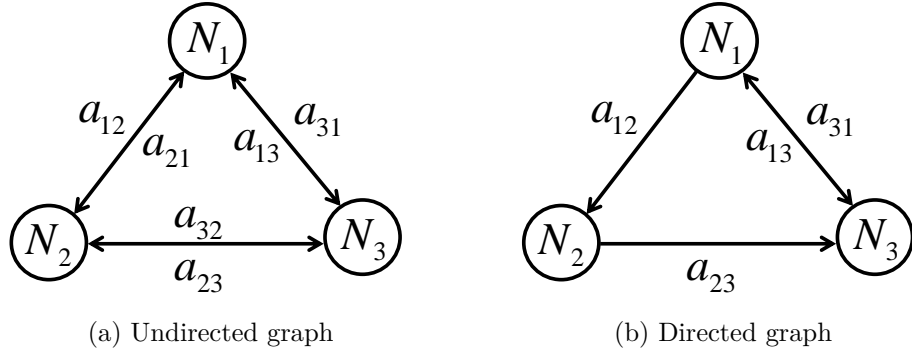


Figure 2.2: Network graphs with different topologies.

Consider a weighted and directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$  with a non-empty set of nodes  $\mathcal{V} = \{1, 2, \dots, N\}$ , a set of edges  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ . An edge rooted at the  $i^{\text{th}}$  node and ended at the  $j^{\text{th}}$  node is denoted by  $(i, j)$ , which means information can flow from the  $i^{\text{th}}$  node to the  $j^{\text{th}}$  node. If an edge exists between nodes  $i$  and  $j$ , we call them *adjacent* and the associated adjacency matrix of the network  $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$ . The weight of edge  $(i, j)$  is  $a_{ij}$  and  $a_{ij} > 0$  if  $(i, j) \in \mathcal{E}$ . The  $j^{\text{th}}$  node is called a neighbour of the  $i^{\text{th}}$  node if  $(j, i) \in \mathcal{E}$ . The in-degree matrix is defined as  $\mathcal{D} = \text{diag}(d_i) \in \mathbb{R}^{N \times N}$  with  $d_i = \sum_{j=1}^N a_{ij}$ . The Laplacian matrix  $\mathcal{L} \in \mathbb{R}^{N \times N}$  of  $\mathcal{G}$  is defined as  $\mathcal{L} = \mathcal{D} - \mathcal{A}$ . For example, the adjacency matrix, in-degree matrix and Laplacian for the directed

topology in Figure 2.2 are given as,

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathcal{L} = \begin{bmatrix} 2 & -1 & -1 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}. \quad (2.12)$$

Suppose a multi-agent network contains both leader and follower agents, and that the information flow among the follower agents is bidirectional and there exists at least one directional link from the leader to the followers. If the  $i^{\text{th}}$  follower agent is connected to the leader (considered as the root node labelled with ‘0’), then an edge  $(0, i)$  is said to exist between them with a gain  $g_i > 0$ . Here the reference trajectory from the root node is applied to the  $i^{\text{th}}$  follower agent which is pinned and hence  $g_i$  is called the pinning gain. The pinning matrix is denoted as  $G = \text{diag}(g_i) \in \mathbb{R}^{N \times N}$ . The convergence of the other follower agents to the reference trajectory is achieved due to their interaction with the pinned agent.

## 2.4 Homogeneous multi-agent systems

This section provides a formal definition of homogeneous multi-agent systems which are considered in this thesis. When the agents of a network are described by identical dynamics, such multi-agent network is said to be homogeneous.

**Definition 2.15** ([68]). A homogeneous multi-agent system is a network of identical multi-input multi-output agents described by

$$\begin{aligned} \text{Agent } i : \dot{x}_i &= Ax_i + Bu_i + Hw_i, \\ y_i &= Cx_i \end{aligned} \quad (2.13)$$

where  $i = \{1, 2, \dots, N\} \in \mathcal{G}$ ,  $x_i \in \mathbb{R}^n$  is the state,  $u_i \in \mathbb{R}^m$  is the input and  $w_i \in \mathbb{R}^w$  is external disturbance.

## 2.5 Negative Imaginary Systems

This section provides the relevant background of NI and SNI systems and some of the internal stability results that exist in the literature. The results here are used later to develop an identification scheme for NI systems and to design an NI based consensus control scheme for a network of tricopters.

**Definition 2.16** (NI systems, [54, 69]). Let  $M(s)$  be the real, rational and proper transfer matrix of a square and causal system without any poles in the open right-half plane.  $M(s)$  is said to be Negative Imaginary (NI) if

- $j [M(j\omega) - M(j\omega)^*] \geq 0$  for all  $\omega \in (0, \infty)$  except the values of  $\omega$  where  $s = j\omega$  is a pole of  $M(s)$ ;
- If  $s = j\omega_0$  with  $\omega_0 \in (0, \infty)$  is a pole of  $M(s)$ , then it is at most a simple pole and the residue matrix  $\lim_{s \rightarrow j\omega_0} (s - j\omega_0)jM(s)$  is Hermitian and positive semidefinite;
- If  $s = 0$  is a pole of  $M(s)$ , then the  $\lim_{s \rightarrow 0} s^k M(s) = 0$  for all  $k \geq 3$  and  $\lim_{s \rightarrow 0} s^2 M(s)$  is Hermitian and positive semidefinite.

A subset of NI systems termed SNI systems is defined as follows:

**Definition 2.17** (SNI systems, [47]). Let  $M(s)$  be the real, rational and proper transfer matrix of a square and causal system.  $M(s)$  is said to be Strictly Negative Imaginary if  $M(s)$  has no poles in  $\{s \in \mathbb{C} : \Re[s] \geq 0\}$  and  $j [M(j\omega) - M(j\omega)^*] \geq 0$  for all  $\omega \in (0, \infty)$ .

### 2.5.1 Properties of networked NI and SNI systems

This subsection declares the properties of the interaction topology of a group of NI systems when considered as agents in a network. These results will be used later to develop an NI-based co-operative control scheme for a network of tricopters.

**Assumption 2.1.** *The communication topology of  $N$  agents (in the homogeneous case) is described by an undirected and connected graph  $\mathcal{G}$ . There always exists a root node (also called the leader or target) which provides reference trajectory to the follower agents (at least to one follower).*



According to Assumption 2.1, we have  $(\mathcal{L}+G) > 0$  where  $G = \text{diag}\{g_1, g_2, \dots, g_N\} > 0$  is the pinning-gain matrix. The following lemma proves that a network of homogeneous LTI systems that satisfies Assumption 2.1 exhibits stable NI (resp. SNI) property if and only if each individual system of the network is stable NI (resp. SNI).

**Lemma 2.5** ([42, 70]). *Consider a network of  $N$  identical stable NI (including SNI) agents that satisfies Assumption 2.1. Then,  $\bar{M}(s) = (\mathcal{L} + G) \otimes M(s)$  is stable NI (resp. SNI) if and only if  $M(s) \in \mathcal{RH}_\infty^{m \times m}$  is NI (resp. SNI).*

Lemma 2.6 shows that a network of all homogeneous stable NI (including SNI) systems retains the same sign definiteness of its DC-gain matrix when the corresponding communication topology satisfies Assumption 2.1.

**Lemma 2.6** ([42, 70]). *Consider a network of  $N$  identical stable NI agents  $M(s) \in \mathcal{RH}_\infty^{m \times m}$  satisfying Assumption 2.1. Denote  $\bar{M}(s) = (\mathcal{L} + G) \otimes M(s)$ . Then,  $\bar{M}(0) > 0$  (resp.  $< 0$ ) if and only if  $M(0) > 0$  (resp.  $< 0$ ).*

### 2.5.2 Eigenvalue loci theory

Similar to a Nyquist plot, the eigenvalue loci  $\rho_i(s)$  for  $i \in \{1, 2, \dots, n\}$  of a transfer function matrix  $G(s)$  is a conformal mapping of the function  $\det[G(s)]$  in a complex plane, known as the eigenvalue loci plane, when  $s$  traverses along the  $s$ -plane  $D$ -contour in the clockwise direction as shown in Figure 2.3. For complete details of the eigenvalue loci theory, please see [71, 72].

The eigenvalue loci theory is used later to prove the stability of a feedback interconnection of NI and VSP controllers.

**Theorem 2.2** ([71, 72]). *The negative feedback interconnection of a plant  $M(s)$  and a controller  $K(s)$  is asymptotically stable if and only if the net sum of the critical point  $(-1 + j0)$  encirclements of all the eigenvalue loci  $\rho_i(j\omega)$  of the loop transfer function  $M(s)K(s)$  for  $i \in \{1, 2, \dots, n\}$  is counter-clockwise and equal to the number of RHP zeros of the open-loop characteristic polynomial. For open-loop stable cases, none of  $\rho_i(j\omega)$  should encircle the critical point  $(-1 + j0)$ .*

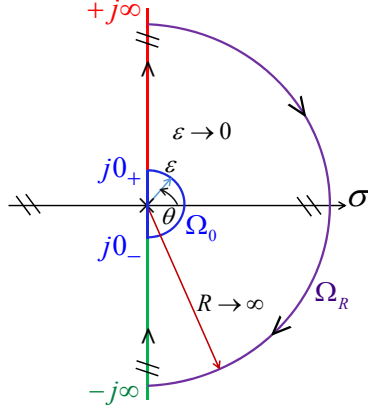


Figure 2.3: Nyquist  $\mathcal{D}$ -contour in the  $s$ -plane, where  $\omega = 0_+, 0_-$  and  $\omega = +\infty, -\infty$  are denoted by  $j0_+, j0_-$  and  $+j\infty, -j\infty$  respectively,  $\Omega_0 = \{s \mid s = \varepsilon e^{j\theta}, \varepsilon \in \mathbb{R}_{>0}, \varepsilon \rightarrow 0, -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}\}$ ,  $\Omega_R = \{s \mid s = R e^{j\theta}, R \in \mathbb{R}_{>0}, R \rightarrow +\infty, -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}\}$  and  $\sigma \in (0, \sigma^*)$  is a finite range.

### 2.5.3 Properties of eigenvalue loci of networked NI (SNI) systems

So far in the literature, the eigenvalue loci theory [71, 72] has been defined for a single LTI system. Lemma 2.7 given below shows that all the eigenvalue loci of networked stable NI (resp. SNI) systems lie below (resp. strictly below) the real axis of the eigenvalue loci plane for all  $\omega \in (0, \infty)$ . This resembles the well-known Nyquist interpretation of the NI and SNI transfer functions in the SISO setting [47].

**Lemma 2.7** ([42, 70]). *Consider a network of  $N$  identical stable NI (resp. SNI) agents  $M(s) \in \mathcal{RH}_\infty^{m \times m}$  satisfying Assumption 2.1. Denote  $\bar{M}(s) = (\mathcal{L} + G) \otimes M(s)$ . Then, the eigenvalue loci  $\rho_i(j\omega)$  of  $\bar{M}(s) \in \mathcal{RH}_\infty^{Nm \times Nm}$  lie below (resp. strictly below) the real axis of the eigenvalue loci plane  $\forall \omega \in (0, \infty)$  and  $\forall i \in \{1, 2, \dots, Nm\}$ .*

The following lemma, referred to as the NI lemma, provides a state-space characterisation for NI systems without poles at the origin.

**Lemma 2.8** (NI Lemma, [47, 73]). *Let  $G(s)$  be the real, rational and proper transfer function matrix of a finite-dimensional, square and causal system  $G$  having a minimal state-space realization  $\left[ \begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]$ . Then,  $G(s)$  is NI without poles at the origin if and*

only if  $\det(A) \neq 0$ ,  $D = D^\top$  and there exists a real matrix  $X = X^\top > 0$  such that

$$\begin{aligned} AX + XA^\top &\leq 0 \quad \text{and} \\ B &= -AXC^\top. \end{aligned} \tag{2.14}$$

## 2.6 Quaternion Maths

Quaternions are useful in modelling the attitude of rotating bodies such as UAVs and a brief introduction to quaternions is given in this section. Let  $\mathbf{q}_v = \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 \in \mathbb{R}^3$ , and  $q_0$  be a scalar, then a quaternion  $\mathbf{q} \in \mathbb{R}^4$  is the sum [74],

$$\mathbf{q} = q_0 + \mathbf{q}_v = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 \tag{2.15}$$

where  $q_i$  for  $i \in \{0, 1, 2, 3\}$  are the components of the quaternion. The quaternion may be represented as a vector

$$\mathbf{q} = [q_0 \ q_1 \ q_2 \ q_3]^\top = \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix}. \tag{2.16}$$

The conjugate, norm and inverse of the quaternion are [74],

$$\mathbf{q}^* = [q_0 \ -q_1 \ -q_2 \ -q_3]^\top, \tag{2.17}$$

$$\|\mathbf{q}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}, \tag{2.18}$$

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|}. \tag{2.19}$$

The product of two quaternions  $\mathbf{p} = p_0 + \mathbf{p}_v$  and  $\mathbf{q} = q_0 + \mathbf{q}_v$  is given as [74],

$$\mathbf{r} = \mathbf{p} \circ \mathbf{q} = p_0q_0 - \mathbf{p}_v \cdot \mathbf{q}_v + p_0\mathbf{q}_v + q_0\mathbf{p}_v + \mathbf{p}_v \times \mathbf{q}_v \tag{2.20}$$

where  $\mathbf{r}$  is itself a quaternion. More compactly, this product may be written as,

$$\mathbf{r} = \mathbf{p} \circ \mathbf{q} = \mathbf{Q}(\mathbf{p})\mathbf{q} = \bar{\mathbf{Q}}(\mathbf{q})\mathbf{p} \tag{2.21}$$

where  $\mathbf{Q} \in \mathbb{R}^{4 \times 4}$ ,  $\bar{\mathbf{Q}} \in \mathbb{R}^{4 \times 4}$  are matrix-valued functions which we define as,

$$\mathbf{Q}(\mathbf{p}) = \begin{bmatrix} p_0 & -\mathbf{p}_v^\top \\ \mathbf{p}_v & p_0\mathbf{I}_3 + \mathbf{S}(\mathbf{p}_v) \end{bmatrix} = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix}, \tag{2.22}$$

and

$$\bar{\mathbf{Q}}(\mathbf{q}) = \begin{bmatrix} q_0 & -\mathbf{q}_v^\top \\ \mathbf{q}_v & p_0 \mathbf{I}_3 - \mathbf{S}(\mathbf{q}_v) \end{bmatrix} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \quad (2.23)$$

where  $\mathbf{S}(\cdot) \in \mathbb{R}^{3 \times 3}$  is the skew-symmetric cross product matrix function as given by (2.1) and this property makes the quaternion product non-commutative as evident from the lower-right  $(3 \times 3)$  sub-matrix of (2.22) and (2.23) respectively. For completeness, the functions  $\mathbf{Q}(\mathbf{q})$  and  $\bar{\mathbf{Q}}(\mathbf{p})$  may be defined similar to (2.22) and (2.23), respectively. Furthermore, by substituting (2.17) into  $\mathbf{Q}(\mathbf{q})$  and  $\bar{\mathbf{Q}}(\mathbf{q})$ , we find that

$$\mathbf{Q}(\mathbf{q}^*) = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix} = \mathbf{Q}(\mathbf{q})^\top \quad (2.24)$$

and

$$\bar{\mathbf{Q}}(\mathbf{q}^*) = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix} = \bar{\mathbf{Q}}(\mathbf{q})^\top. \quad (2.25)$$

Finally, it is clear from (2.19) that when the norm is unity,  $\mathbf{q}^{-1} = \mathbf{q}^*$  and such quaternion is called a *unit quaternion*. For the rest of this thesis, we assume unit quaternions.

## 2.7 Coordinate Rotations and Rotation matrices

The dynamics of a rigid-body may be represented in different coordinate systems with the possibility of moving from one coordinate system to another. This section provides a background of coordinate systems and rotation matrices and the results discussed here are used later in deriving the dynamic equations of the tricopter. Consider Figure 2.4(a), with origin 0,  $x$  and  $y$  axes, and  $\alpha$ , the angle between the vector  $\mathbf{v}$  and the positive  $x$  axis:

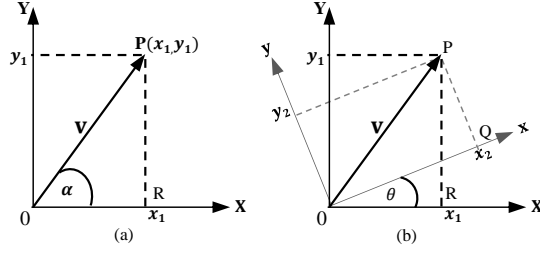


Figure 2.4: Rotation of coordinates

Any point  $P$  can be located by an ordered pair  $(x_1, y_1)$  to yield a  $\mathbb{R}^2$  coordinate plane. Figure 2.4(b) shows the vector  $\mathbf{v}$  after a rotation of the coordinate system through an angle  $\theta$  about the origin. Using Figure 2.4, it can be shown that the rotation about the origin of a  $\mathbb{R}^2$  coordinate system in matrix form is given as [74],

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}. \quad (2.26)$$

The  $\mathbb{R}^2$  coordinate plane in Figure 2.4 may be extended to a  $\mathbb{R}^3$  coordinate plane using a third  $z$ -axis which is assumed to be perpendicular to the  $xy$  plane and directed positively out of the paper towards the reader, forming a right-handed system [74].

Thus, let  $\mathbf{v}$  have coordinates  $(x_1, y_1, z_1)$ . If the coordinate frame is rotated about the  $z$ -axis through the angle  $\theta$  as in Figure 2.4(b), such that the new coordinates of the rotated frame are  $(x_2, y_2, z_2)$ , it is clear that the rotation about the  $z$ -axis will not change the  $z$ -coordinate [74]. That is,  $z_2 = z_1$ . Combining this result with (2.26) yields,

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \triangleq \mathbf{v}_2 = \mathbf{R}_z \mathbf{v}_1 \quad (2.27)$$

where  $\mathbf{R}_z$  is called a *rotation matrix* and describes a rotation about the  $z$ -axis mapping from coordinate frame 1  $(x_1, y_1, z_1)$  to 2  $(x_2, y_2, z_2)$ . A similar argument holds for the rotation of the coordinate frame through the angle  $\theta$  about the  $x$ -axis ( $x_2 = x_1$ ) and  $y$ -axis ( $y_2 = y_1$ ) so that

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \text{ and } \mathbf{R}_y = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix},$$

respectively. The matrices  $\mathbf{R}_x$ ,  $\mathbf{R}_y$  and  $\mathbf{R}_z$  will be used later to derive Euler rotation sequences.

### 2.7.1 Rotation matrices

A rotation matrix is one which when multiplied with a vector rotates the vector while preserving its length. The *special orthogonal group* of all rotations in 3 dimension is denoted by  $\mathcal{SO}(3)$ . The matrices in this group satisfy  $A^{-1} = A^\top$  hence the term orthogonal. This group is termed *special* because the determinant of its matrices is 1. In short, if  $A \in \mathcal{SO}(3)$ ,  $\det(A) = 1$  and  $A^{-1} = A^\top$ . Hence, let  $\mathbf{R}_b^a \in \mathcal{SO}(3)$  be a rotation matrix that transforms from coordinate system  $b$  to  $a$ , then the following properties hold:

$$(\mathbf{R}_b^a)^{-1} = (\mathbf{R}_b^a)^\top = \mathbf{R}_a^b, \quad (2.28)$$

$$\mathbf{R}_b^c \mathbf{R}_a^b = \mathbf{R}_a^c, \quad (2.29)$$

$$\det[\mathbf{R}_a^b] = 1. \quad (2.30)$$

Consequently, a vector in the earth-fixed inertial frame can be transformed to the body-fixed coordinate frame by pre-multiplying it by a rotation matrix as described by the following definition.

**Fact 2.1.** Let  $\mathbf{v}^e \in \mathbb{R}^3$  be a vector in the earth-fixed inertial coordinate frame, and  $\mathbf{v}^b \in \mathbb{R}^3$  be the same vector in body-fixed coordinates, then the following relations hold:

$$\mathbf{v}^b = \mathbf{R}_e^b \mathbf{v}^e, \quad (2.31)$$

$$\mathbf{v}^e = (\mathbf{R}_e^b)^\top \mathbf{v}^b. \quad (2.32)$$

### 2.7.2 Quaternion rotation operators

A unit quaternion may be used to represent the attitude of a rigid body and unit quaternions are used to represent the attitude of the multi-tilt tricopter in this thesis. Let us define a unit vector  $\hat{\mathbf{u}} = \frac{\mathbf{q}_v}{\sin(\theta/2)}$ , which represents the direction of  $\mathbf{q}$  and  $\theta$  is the angle associated with the quaternion.

**Theorem 2.3.** [74, Theorem 5.2] For any unit quaternion

$$\mathbf{q} = q_0 + \mathbf{q}_v = \cos \frac{\theta}{2} + \hat{\mathbf{u}} \sin \frac{\theta}{2} \quad (2.33)$$

and for any vector  $\mathbf{v} \in \mathbb{R}^3$ , the action of the operator

$$L_{\mathbf{q}^*}(\mathbf{v}) = \mathbf{q}^* \circ \mathbf{v} \circ \mathbf{q} \quad (2.34)$$

is a rotation of the coordinate frame about the axis  $\hat{\mathbf{u}}$  through an angle  $\theta$  while  $\mathbf{v}$  is not rotated.

Thus, consider a vector  $\mathbf{v}^e \in \mathbb{R}^3$  in earth coordinates. If  $\mathbf{v}^b \in \mathbb{R}^3$  is the same vector in body coordinates then,

$$\begin{bmatrix} 0 \\ \mathbf{v}^b \end{bmatrix} = \mathbf{q}^* \circ \begin{bmatrix} 0 \\ \mathbf{v}^e \end{bmatrix} \circ \mathbf{q} = \bar{\mathbf{Q}}(\mathbf{q}) \mathbf{Q}(\mathbf{q})^\top \begin{bmatrix} 0 \\ \mathbf{v}^e \end{bmatrix}, \quad (2.35)$$

$$= \begin{bmatrix} 1 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{R}_e^b(\mathbf{q}) \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{v}^e \end{bmatrix} \quad (2.36)$$

where

$$\mathbf{R}_e^b(\mathbf{q}) = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}. \quad (2.37)$$

It can also be shown that the angular velocity  $\boldsymbol{\omega}^b$  is related to the unit quaternion through the following relation [24, 74],

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \circ \begin{bmatrix} 0 \\ \boldsymbol{\omega}^b \end{bmatrix}. \quad (2.38)$$

### 2.7.3 Euler angles and rotation sequences

The orientation of one coordinate system with respect to another can be described by three successive coordinate rotations around the orthogonal coordinate axes, and the angles of rotation are called the *Euler angles* [24, 63, 74]. Euler angles are used in this thesis to represent the attitude of the tricopters.

**Theorem 2.4** ([74]). *Any two independent orthonormal coordinate frames can be related by a sequence of rotations (not more than three) about coordinate axes, where no two successive rotations may be about the same axis.*

This thesis uses the aerospace  $(z, y, x)$  or  $(3, 2, 1)$  sequence [63, 74] that is required to go from a reference on earth into alignment with an aircraft body-fixed coordinate system. From the reference, the system of rotation is first a rotation through an angle  $\psi$  about the  $z$ -axis (heading), followed by a rotation through an angle  $\theta$  about the new  $y$ -axis (pitch) and finally a rotation through an angle  $\phi$  about the new  $x$ -axis (roll) and expressed as the matrix product

$$\begin{aligned} \mathbf{R}_e^b(\boldsymbol{\eta}) &= \mathbf{R}_x(\phi) \mathbf{R}_y(\theta) \mathbf{R}_z(\psi), \\ &= \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \end{aligned} \quad (2.39)$$

where  $c_\phi \triangleq \cos \phi$ ,  $s_\phi \triangleq \sin \phi$  and  $\boldsymbol{\eta} = [\phi \ \theta \ \psi]^\top$ .

The following definition describes the relationship between the time derivatives of the Euler angles and the angular velocities of a rigid body.

**Definition 2.18** ([24]). Let  $\dot{\boldsymbol{\eta}} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^\top$  be the time derivative of the Euler angles of a rigid body, the relationship between  $\dot{\boldsymbol{\eta}}$  and the angular velocity of the body  $\boldsymbol{\omega}^b = [\omega_p \ \omega_q \ \omega_r]^\top$  is given by,

$$\boldsymbol{\omega}^b = \boldsymbol{\Gamma}(\boldsymbol{\eta}) \dot{\boldsymbol{\eta}} \quad (2.40)$$

where  $\boldsymbol{\Gamma}(\boldsymbol{\eta}) = [\hat{\mathbf{u}}_i, \mathbf{R}_x(\phi) \hat{\mathbf{u}}_j, \mathbf{R}_x(\phi) \mathbf{R}_y(\theta) \hat{\mathbf{u}}_k]$  is an *Euler angle rates matrix* function with  $\hat{\mathbf{u}}_i, \hat{\mathbf{u}}_j, \hat{\mathbf{u}}_k$  as the  $i^{th}, j^{th}$  and  $k^{th}$  unit vectors respectively such that

$$\boldsymbol{\Gamma}(\boldsymbol{\eta}) = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \theta & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix}. \quad (2.41)$$



### 2.7.4 Relationship between Quaternions and Euler angles

It is often convenient to transform vectors from a quaternion representation to euler angles. The Euler angles may be recovered from (2.37) as [74],

$$\boldsymbol{\eta} = \begin{bmatrix} \text{atan2}(2q_2q_3 + 2q_0q_1, 1 - 2(q_1^2 + q_2^2)) \\ -\text{asin}(2q_1q_3 - 2q_0q_2) \\ \text{atan2}(2q_1q_2 + 2q_0q_3, 1 - 2(q_2^2 + q_3^2)) \end{bmatrix}. \quad (2.42)$$

## 2.8 Summary

This chapter has provided a brief but necessary overview of the key concepts, techniques and tools which are used in different parts of this work. The earlier sections covered linear algebra, linear systems theory, graph theory and negative imaginary systems theory. The later sections described coordinate systems, representations and transformations which are useful in modelling rigid bodies. Overall, the material presented is important for the reader to effectively understand the remaining chapters of this work as several references are made to this chapter throughout this thesis.

## Chapter 3

# Development and stabilization of a low-cost single-tilt tricopter

All the material in this chapter was published in [75].

### 3.1 Introduction

In this chapter, a single-tilting tricopter using low-cost materials and open-source software with optical flow included for GPS-denied environments is constructed. Although researchers have paid some attention to the tricopter UAV, there is still insufficient experimental research to validate the tricopter concept onto real physical hardware. Most of the existing works have predominantly focused on theoretical and simulation results with little or no consideration of practicability. Also, only few of the models used in literature are based on real experimental data collected from physical hardware, the rest are arbitrarily chosen parameters used as numerical examples. These issues open up opportunities for further research and this is the motivation for this work. In this chapter, the dynamic model of the tricopter is derived, an experiment to obtain the actuator constants from the acquired data is developed together with a CAD model based on the measured parameters of the tricopter, which is used for estimating the moments of inertia. A control allocation scheme is proposed which allocates the actuator signals via a non-square mixer matrix due to under-actuation in the tricopter. The feasibility and applicability of the methods used herein is shown by implementing cascaded-PID control to stabilize the tricopter, and validated with some

trial experiments on the developed platform. Furthermore, this work serves as a basis for more complex control and real-time hardware experiments with varied scenarios on the developed tricopter which is the end goal of this project.

The key contribution to the state-of-the art from this chapter is:

- The development of control allocation scheme which allocates the actuator signals via a non-square mixer matrix due to under-actuation in the tricopter.

The other technical contributions of this chapter are summarized as follows:

- Development of a single-tilting tricopter using low-cost materials and open-source software with optical flow included for GPS-denied environments,
- The dynamic model is derived, an experiment to obtain actuator constants from acquired data is developed, and a CAD model is drafted from measured parameters and used for estimating moment of inertia,
- Implementation of cascaded-PID control to stabilize the single-tilt tricopter UAV, simulated with Matlab/Simulink and validated with trial experiments on the developed hardware.

## 3.2 Mathematical Modeling

Besides the arrangement of the rotors and choice of coordinate system, the dynamic model developed in this section is similar to those in existing literature. However, the parameters used to fit the model are based on the actual hardware information rather than arbitrary parameters. As shown in Figure 3.1,  $(X^e, Y^e, Z^e)$  denotes the earth coordinate system which is assumed to be inertial and  $(X^b, Y^b, Z^b)$  denotes the body coordinate system with its origin fixed to the center of mass  $\mathbf{G}$  of the vehicle. Additionally, Figure 3.2 shows a technical diagram of the top view of the tricopter for convenience. The transformation from the inertial frame to the body frame following the  $(z, y, x)$  sequence [63] is encoded in the rotation matrix  $\mathbf{R}_e^b(\boldsymbol{\eta})$  given by (2.39) and the reverse transformation from body frame to inertial frame is obtained as the inverse  $\mathbf{R}_e^b(\boldsymbol{\eta})^{-1} = \mathbf{R}_e^b(\boldsymbol{\eta})^\top = \mathbf{R}_b^e(\boldsymbol{\eta})$  from rotation matrix properties [63]. Similarly, the function which relates the angular velocities in body coordinates to the euler angle rates is given by (2.41).

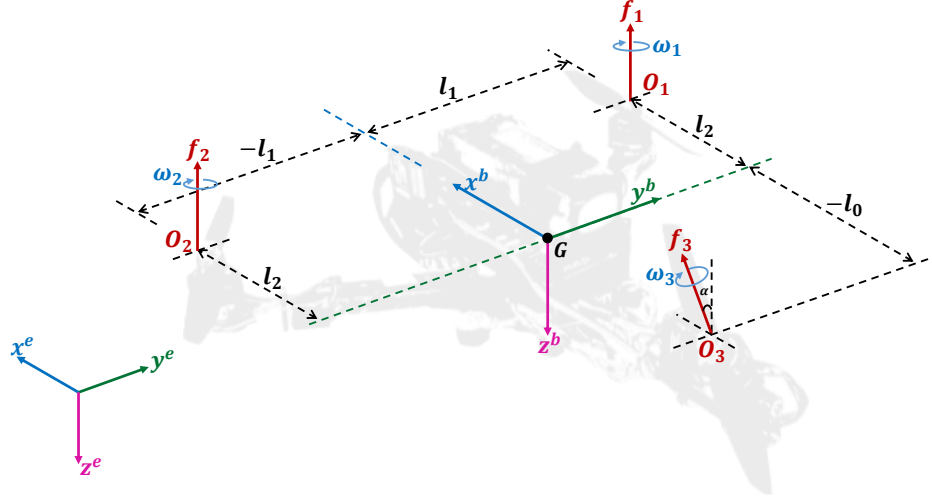


Figure 3.1: Forces and torques acting on single-tilt tricopter and coordinate systems.

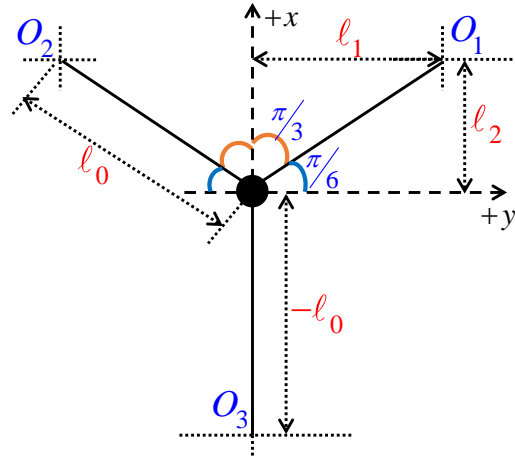


Figure 3.2: Top view diagram of tricopter where  $l_0$  is the length of each arm,  $l_1 = l_0 \sin \frac{\pi}{3}$ ,  $l_2 = l_0 \cos \frac{\pi}{3}$ .

Using this information, let us now derive the forces and torques acting on the single-tilt tricopter.

### 3.2.1 Forces and Torques

The forces and drag torques produced by each rotor as depicted in Figure 3.1 are assumed to be proportional to the square of the angular speeds  $\omega_i$  [76] since the propeller is directly coupled with the DC motor such that

$$f_i = k_t \omega_i^2 \text{ and } \tau_i = k_d \omega_i^2 \quad \forall i \in \{1, 2, 3\} \quad (3.1)$$

where  $f_i$  and  $\tau_i$  denote the forces and drag torques respectively,  $k_t$  is the thrust constant and  $k_d$  is the drag-torque constant. The force produced by the  $i$ th rotor from Figure 3.1 is thus

$$\mathbf{f}_i = \begin{bmatrix} 0 \\ 0 \\ -k_t \omega_i^2 \end{bmatrix} \text{ for } i \in \{1, 2\} \text{ and } \mathbf{f}_3 = \begin{bmatrix} 0 \\ -k_t \omega_3^2 \sin \alpha \\ -k_t \omega_3^2 \cos \alpha \end{bmatrix}$$

so that the total force from all three rotors is given as

$$\mathbf{F}_m^b = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} 0 \\ -k_t \omega_3^2 \sin \alpha \\ -k_t (\omega_1^2 + \omega_2^2 + \omega_3^2 \cos \alpha) \end{bmatrix}. \quad (3.2)$$

Let  $(o_1, o_2, o_3)$  be the application points of the forces  $(\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)$  respectively. Then the torque generated by the rotors with respect to the center of mass  $\mathbf{G}$  can be expressed in the body frame as

$$\mathbf{T}_r^b = \sum_{i=1}^3 \boldsymbol{\tau}_{m,i}^b = \sum_{i=1}^3 (\mathbf{G}_{oi} \times \mathbf{f}_i) \quad (3.3)$$

where  $\mathbf{G}_{oi}$  is the vector of the distance of the  $i^{\text{th}}$  rotor from the center of gravity  $\mathbf{G}$  and

$$\mathbf{G}_{o1} = \begin{bmatrix} l_2 \\ l_1 \\ 0 \end{bmatrix}, \quad \mathbf{G}_{o2} = \begin{bmatrix} l_2 \\ -l_1 \\ 0 \end{bmatrix}, \quad \mathbf{G}_{o3} = \begin{bmatrix} -l_0 \\ 0 \\ 0 \end{bmatrix}$$

where  $l_0$  is the length of each rotor arm measured from each of the rotor heads to the center of mass  $\mathbf{G}$ ,  $l_1 = l_0 \sin \frac{\pi}{3}$  and  $l_2 = l_0 \cos \frac{\pi}{3}$ . Applying these in (3.3), the torque produced by the rotors is

$$\mathbf{T}_r^b = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} l_1 k_t (\omega_2^2 - \omega_1^2) \\ l_2 k_t (\omega_1^2 + \omega_2^2) - l_0 k_t \omega_3^2 \cos \alpha \\ l_0 k_t \omega_3^2 \sin \alpha \end{bmatrix}. \quad (3.4)$$

The drag torque on the propellers is opposite to the direction of rotation of the propellers. From Figure 3.1, the reaction torques of the  $i$ th rotor are given as

$$\boldsymbol{\tau}_{d,i}^b = \begin{bmatrix} 0 \\ 0 \\ -k_d \omega_i^2 \end{bmatrix} \quad \forall i \in \{1, 2\} \quad \text{and} \quad \boldsymbol{\tau}_{d,3}^b = -k_d \omega_3^2 \begin{bmatrix} 0 \\ \sin \alpha \\ \cos \alpha \end{bmatrix}, \quad (3.5)$$

so that the total drag torque from all three rotors is

$$\mathbf{T}_d^b = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 0 \\ -k_d \omega_3^2 \sin \alpha \\ -k_d(\omega_1^2 + \omega_2^2 + \omega_3^2 \cos \alpha) \end{bmatrix}. \quad (3.6)$$

By summing (3.4) and (3.6) and grouping the result with (3.2), the expression for the total forces and torques which describes the mixer of the tricopter is obtained as

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 0 \\ -k_t \omega_3^2 \sin \alpha \\ -k_t(\omega_1^2 + \omega_2^2 + \omega_3^2 \cos \alpha) \\ l_1 k_t(\omega_2^2 - \omega_1^2) \\ l_2 k_t(\omega_1^2 + \omega_2^2) - l_0 k_t \omega_3^2 \cos \alpha - k_d \omega_3^2 \sin \alpha \\ l_0 k_t \omega_3^2 \sin \alpha - k_d(\omega_1^2 + \omega_2^2 + \omega_3^2 \cos \alpha) \end{bmatrix}. \quad (3.7)$$

### 3.2.2 Newton-Euler Model

The tricopter UAV is considered to be a rigid-body with mass  $m_t$ , and the total force acting on the UAV  $\mathbf{F}_t^b$  is the sum of the force produced by the rotors  $\mathbf{F}_m^b$  and the force due to gravity

$$\mathbf{F}_g^e = \begin{bmatrix} 0 \\ 0 \\ m_t g \end{bmatrix} \quad (3.8)$$

where  $g$  is the acceleration due to gravity. Then, define  $\boldsymbol{\xi}^e \triangleq [x \ y \ z]^\top$ ,  $\boldsymbol{\eta}^b \triangleq [\phi \ \theta \ \psi]^\top$ ,  $\boldsymbol{\omega}^b \triangleq [\omega_p \ \omega_q \ \omega_r]^\top$ . By using Newton-Euler methods [63], the translational dynamics of the single-tilt tricopter is,

$$\ddot{\boldsymbol{\xi}}^e = \frac{1}{m_t} [\mathbf{R}_e^b(\boldsymbol{\eta})^\top \mathbf{F}_m^b + \mathbf{F}_g^e], \quad (3.9)$$

and the rotational dynamics from (2.40) is given by,

$$\dot{\boldsymbol{\eta}} = \boldsymbol{\Gamma}(\boldsymbol{\eta})^{-1} \boldsymbol{\omega}^b. \quad (3.10)$$

The angular accelerations are given as,

$$\dot{\boldsymbol{\omega}}^b = \mathbf{J}^{-1} \left[ (-\boldsymbol{\omega}^b \times \mathbf{J} \boldsymbol{\omega}^b) + \mathbf{T}_m^b \right] \quad (3.11)$$

where  $\mathbf{T}_m^b = \mathbf{T}_r^b + \mathbf{T}_d^b$  is the total torque applied to the tricopter and  $\mathbf{J}$  is the inertia matrix. Equations (3.9)–(3.11) together with (3.7) describe the nonlinear model of the tricopter. Note that there is no particular reason for using these equations in this form with the translational and rotational components in different frames as these equations may be put in the same frame. The interested reader is also referred to [20] which describes a model where all three rotors can tilt. The model in [20] is an extension of the classic tricopter's tail rotor tilt action to the other two rotors.

## 3.3 Model Parameters

### 3.3.1 Moments of Inertia

A CAD model was developed in Solidworks as depicted in Figure 3.3 using the manually measured parameters of the tricopter and this was used to obtain the moments of inertia. It is assumed that the fuselage is a cuboid with length  $a$ , breadth  $b$ , height  $c$  and mass  $m_0$ , and that the motors are cylindrical with diameter  $D$ , height  $h$  and mass  $m_1$ . Note that  $l_1 = \frac{\sqrt{3}}{2}l_0$  and  $l_2 = \frac{1}{2}l_0$ .

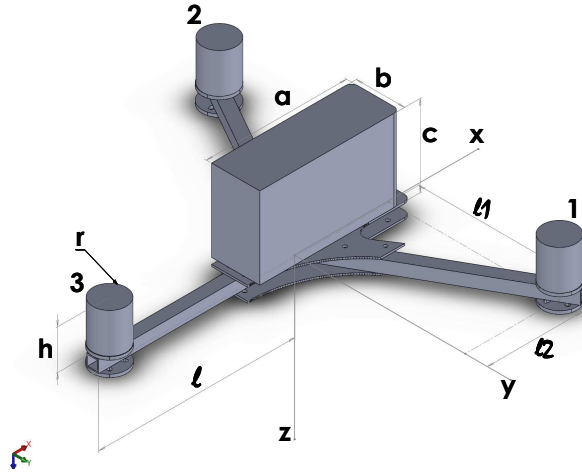


Figure 3.3: Derivation of moments of inertia of a tricopter. Note that  $l = l_0$ .

The components  $J_{xy}$ ,  $J_{xz}$  and  $J_{yz}$  are small compared to the others and are assumed negligible so that the inertia matrix becomes  $\mathbf{J} = \text{diag}\{J_{xx}, J_{yy}, J_{zz}\}$ . For convenience, let  $J_{xx} \triangleq J_x$ ,  $J_{yy} \triangleq J_y$  and  $J_{zz} \triangleq J_z$ . The nominal moments of inertia are given in Table 3.1.

### 3.3.2 Thrust and Torque Constants

The thrust (and torque) at different speeds were measured through an experiment which consists of a thrust stand and dynamometer. The stand has a load cell for measuring the thrust, and two adjacent load cells for measuring torque. The motor is mounted between these load cells and the torque is measured using a pivot system by computing the moment between these two load cells. This arrangement is depicted in Figure 3.4.

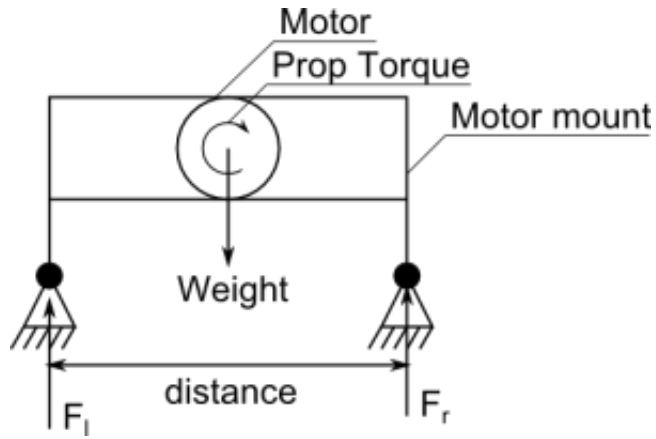


Figure 3.4: Freebody diagram of thrust stand

The stand was fitted to a wooden board mounted on a bench as in Figure 3.5.

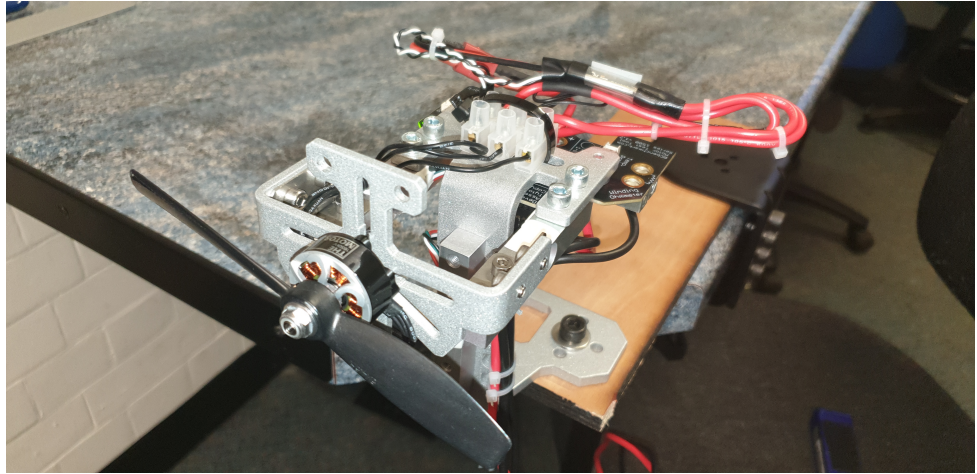


Figure 3.5: Measurement of thrust and torque constants.

The thrust constant  $k_t$  and drag-torque constant  $k_d$  are obtained by plotting thrust and drag torque against the square of the speed respectively so that the constants  $k_t$  and  $k_d$  are simple gradients of the best fitting line through all the data points, constructed via least squares. The experimental data used to obtain the constants  $k_t$



and  $k_d$  are shown in Figure 3.6 from which we obtain  $k_t = 1.591 \times 10^{-6}$  N/(rad/s)<sup>2</sup> and  $k_d = 2.354 \times 10^{-8}$  N.m/(rad/s)<sup>2</sup> when using Emax2207-eco motors with  $6 \times 4.5$  propellers (6 inches long with a pitch angle of 4.5 degrees) on 3S (11.10 Volts). All

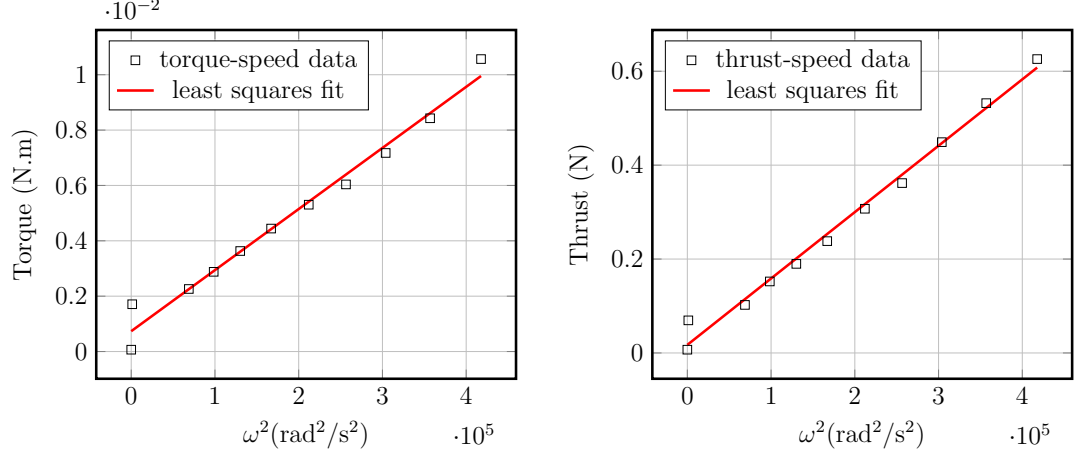


Figure 3.6: Thrust and drag-torque constants data with regression fit.

the parameters for the single-tilt tricopter system including the moments of inertia, are given in Table 3.1. It should be noted that the results in Figure 3.6 were obtained using only one of the motors of the tricopter. This implies that there may be some deviations in the value of the torque and thrust constants of the other two motors. However, as the motors are of identical specification, it is assumed that any such deviations are small and negligible.

Table 3.1: Summary of estimated parameters

Parameter	Estimate
arm length, $l_0$	$1.625 \times 10^{-1}$ m
distance of M1 from $\mathbf{G}$ on $y$ -axis, $l_1$	$1.4073 \times 10^{-1}$ m
distance of M1/M2 from $\mathbf{G}$ on $x$ -axis, $l_2$	$8.125 \times 10^{-2}$ m
length of fuselage, $a$	$9.221 \times 10^{-2}$ m
width of fuselage, $b$	$4.968 \times 10^{-2}$ m
height of fuselage, $c$	$8.493 \times 10^{-2}$ m
mass of motor, $m_1$	$4 \times 10^{-2}$ kg
mass of fuselage, $m_0$	$5.83 \times 10^{-1}$ kg
radius of motor, $r$	$1.375 \times 10^{-2}$ m
height of motor, $h$	$3.276 \times 10^{-2}$ m
thrust constant, $k_t$	$1.591 \times 10^{-6}$ kg.m
drag torque constant, $k_d$	$2.354 \times 10^{-8}$ kg.m <sup>2</sup>
moment of inertia in $x$ -axis, $J_x$	$2.33 \times 10^{-3}$ kg.m <sup>2</sup>
moment of inertia in $y$ -axis, $J_y$	$2.71 \times 10^{-3}$ kg.m <sup>2</sup>
moment of inertia in $z$ -axis, $J_z$	$4.36 \times 10^{-3}$ kg.m <sup>2</sup>
moment of inertia in $xy$ -axes, $J_{xy}$	$1.12 \times 10^{-7}$ kg.m <sup>2</sup>
moment of inertia in $xz$ -axes, $J_{xz}$	$-1.0 \times 10^{-5}$ kg.m <sup>2</sup>
moment of inertia in $yz$ -axes, $J_{yz}$	$1.44 \times 10^{-8}$ kg.m <sup>2</sup>

## 3.4 Hover Control design

### 3.4.1 Linearised Model

The main objective is to achieve stability of the tricopter in hover. This is essential since a stable hover implies that all the attitude angles (or channels) are stable. Furthermore, this is a basic requirement to implement trajectory or position tracking control since the attitude loops are the low level (inner) loops while the position controllers are the outer loops. Since we are only interested in operating the tricopter close to hover, we can simplify the nonlinear model of subsection 3.2.2 via linearisation. Thus, only the dynamics that describe the tricopter's behaviour when close to hovering state are considered. This leads to the assumption that  $\phi \approx 0, \theta \approx 0, \psi \approx 0$  so that  $\cos \phi \approx \cos \theta \approx \cos \psi \approx 1$  and  $\sin \phi \approx \phi, \sin \theta \approx \theta, \sin \psi \approx \psi$ . Let the state vectors be defined as  $\mathbf{x} = (x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi})$ . By applying the small angle assumptions in (3.9) and (3.10), and linearising about the operating point  $\bar{\mathbf{x}} = (\bar{x}, \bar{y}, \bar{z}, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ , we obtain the linearised dynamics [77],

$$\begin{cases} \ddot{\phi} = \frac{1}{J_{xx}}\tau_x, & \ddot{\theta} = \frac{1}{J_{yy}}\tau_y, & \ddot{\psi} = \frac{1}{J_{zz}}\tau_z, \\ \ddot{x} = -g\theta, & \ddot{y} = g\phi + \frac{F_y}{m_t}, & \ddot{z} = \frac{F_z}{m_t}. \end{cases} \quad (3.12)$$

### 3.4.2 Control Allocation

It is not straightforward to use the vector (3.7) for control directly due to its complexity, and also because the mixer matrix obtained is non-square due to more control inputs  $[F_y, F_z, \tau_x, \tau_y, \tau_z]^\top$  than actuator signals  $[\omega_1^2, \omega_2^2, \omega_3^2, \alpha]^\top$ . Hence, the actuator signals cannot be computed using an inverse. To solve this problem, the vector (3.7) is split into two groups where  $F_y$  is separated from the main allocation noting that  $F_y$  is due to the tilting angle, and the main lift force is provided by  $F_z$ . The input vector then becomes  $[u_z, u_\phi, u_\theta, u_\psi]^\top = [F_z, \tau_x, \tau_y, \tau_z]^\top$ . The term  $k_d \omega_3^2 \sin \alpha$  in  $\tau_y$  of (3.7) is assumed negligible as  $\alpha$  is small around hover so that the main control allocation is

given as the mixer

$$\begin{bmatrix} u_z \\ u_\phi \\ u_\theta \\ u_\psi \end{bmatrix} = \begin{bmatrix} -k_t & -k_t & -k_t & 0 \\ -l_1 k_t & l_1 k_t & 0 & 0 \\ l_2 k_t & l_2 k_t & -l_0 k_t & 0 \\ -k_d & -k_d & -k_d & l_0 k_t \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \cos \alpha \\ \omega_3^2 \sin \alpha \end{bmatrix} = \mathcal{M} \Omega. \quad (3.13)$$

By performing row reduction on  $\mathcal{M}$ , its row echelon form is obtained as,

$$\mathcal{M}_{\text{REF}} = \begin{matrix} & R2-l_1 R1 \rightarrow \\ & R3+l_2 R1, l_2=\frac{1}{2}l_0 \rightarrow \\ & R4-\frac{k_d}{k_t} R1 \rightarrow \end{matrix} \begin{bmatrix} -k_t & -k_t & -k_t & 0 \\ 0 & 2l_1 k_t & l_1 k_t & 0 \\ 0 & 0 & -\frac{3}{2}l_0 k_t & 0 \\ 0 & 0 & 0 & l_0 k_t \end{bmatrix}. \quad (3.14)$$

It is evident that the matrix  $\mathcal{M}_{\text{REF}}$  is always full rank as long as its diagonal elements are positive and this is true since  $k_t, k_d > 0$  and  $l_i > 0$ ,  $i \in \{0, 1, 2\}$ . Hence, from definitions 2.2 and 2.3,  $\mathcal{M}$  is invertible. By taking the inverse of  $\mathcal{M}$ , vector  $\Omega$  is given by

$$\begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} = \begin{bmatrix} -\frac{l_0}{2k_t(l_0+l_2)}u_z - \frac{1}{2l_1 k_t}u_\phi + \frac{l_0}{2k_t(l_0+l_2)}u_\theta \\ -\frac{l_0}{2k_t(l_0+l_2)}u_z + \frac{1}{2l_1 k_t}u_\phi + \frac{l_0}{2k_t(l_0+l_2)}u_\theta \\ -\frac{l_2}{k_t(l_0+l_2)}u_z - \frac{1}{k_t(l_0+l_2)}u_\theta \\ -\frac{k_d}{k_t^2 l_0}u_z + \frac{1}{k_t}u_\psi \end{bmatrix}. \quad (3.15)$$

Hence,  $\omega_1 = \sqrt{\Omega_1}$ ,  $\omega_2 = \sqrt{\Omega_2}$ ,  $\omega_3 = \sqrt[4]{\Omega_3^2 + \Omega_4^2}$  and  $\alpha = \text{atan}(\frac{\Omega_4}{\Omega_3})$ . Having obtained the speed of rotor 3 as  $\omega_3$  and the tilt angle  $\alpha$  and in order to account for the lateral force which was separated from the main allocation,  $u_y = F_y$  is then allocated based on the  $\omega_3$  and  $\alpha$ , as  $u_y = -k_t \omega_3^2 \sin \alpha$ , in the second allocation.

### 3.4.3 PID Cascade scheme

A PID control scheme was implemented as depicted in Figure 3.7. As an example, the output of the PID controller for the roll rate loop is given as

$$u_\phi = k_P(p_d - p) + k_I \int_0^t (p_d - p) - k_D p \quad (3.16)$$

where  $k_P, k_I, k_D$  are the gains of the PID controller,  $p_d$  is the desired roll rate,  $p$  is the measured roll rate and  $(p_d - p)$  is the error. The derivative gain was applied to the output rather than the error to avoid *derivative kick*.

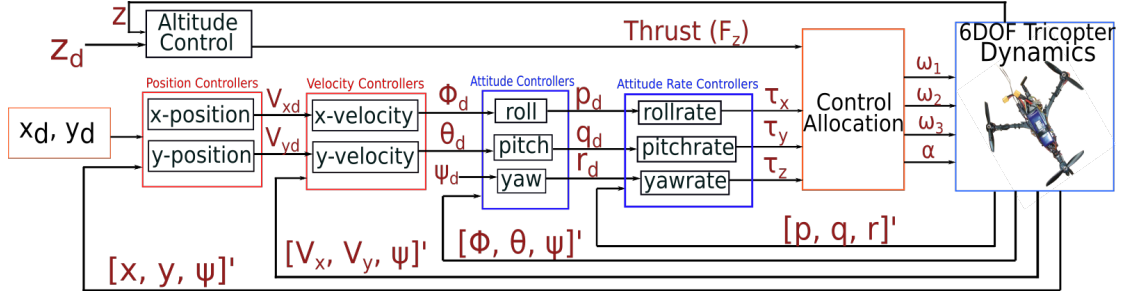


Figure 3.7: Cascaded-PID control architecture.

Independent controllers similar to (3.16) were tuned for the angular rates, the attitude, the linear velocities and the positions respectively using Simulink with a criteria of settling time  $t_s = 6$  seconds. The difficulty of tuning such a cascade structure shown in Figure 3.7 is that the outer loops (or higher level loops) depend on the inner loops. Hence, the lower level loops were tuned first starting with the attitude rate loop and then extending to the attitude loop and so on. As the tricopter is unstable in open loop, methods such as Ziegler-Nichols step response method [78] for tuning PID controllers could not be utilized. The Ziegler-Nichols closed loop method was also attempted but it was difficult to obtain a finite value for the ultimate gain  $K_u$ . Consequently, the PID tuner in Simulink Control Design was used to tune each of the loops separately. The final PID gains after some fine-tuning are summarized in Tables 3.2 to 3.5.

Table 3.2: PID gains for attitude rate loop

	$(p, p_d) \rightarrow u_\phi$	$(q, q_d) \rightarrow u_\theta$	$(r, r_d) \rightarrow u_\psi$
$k_P$	0.019	0.025	0.093
$k_I$	0.018	0	0.139
$k_D$	0	0	0

Table 3.3: PID gains for attitude loop

	$(\phi, \phi_d) \rightarrow p_d$	$(\theta, \theta_d) \rightarrow q_d$	$(\psi, \psi_d) \rightarrow r_d$
$k_P$	4.75	4.75	4.75
$k_I$	3.85	3.83	3.85
$k_D$	0.74	0.74	0.74

Table 3.4: PID gains for velocity loop

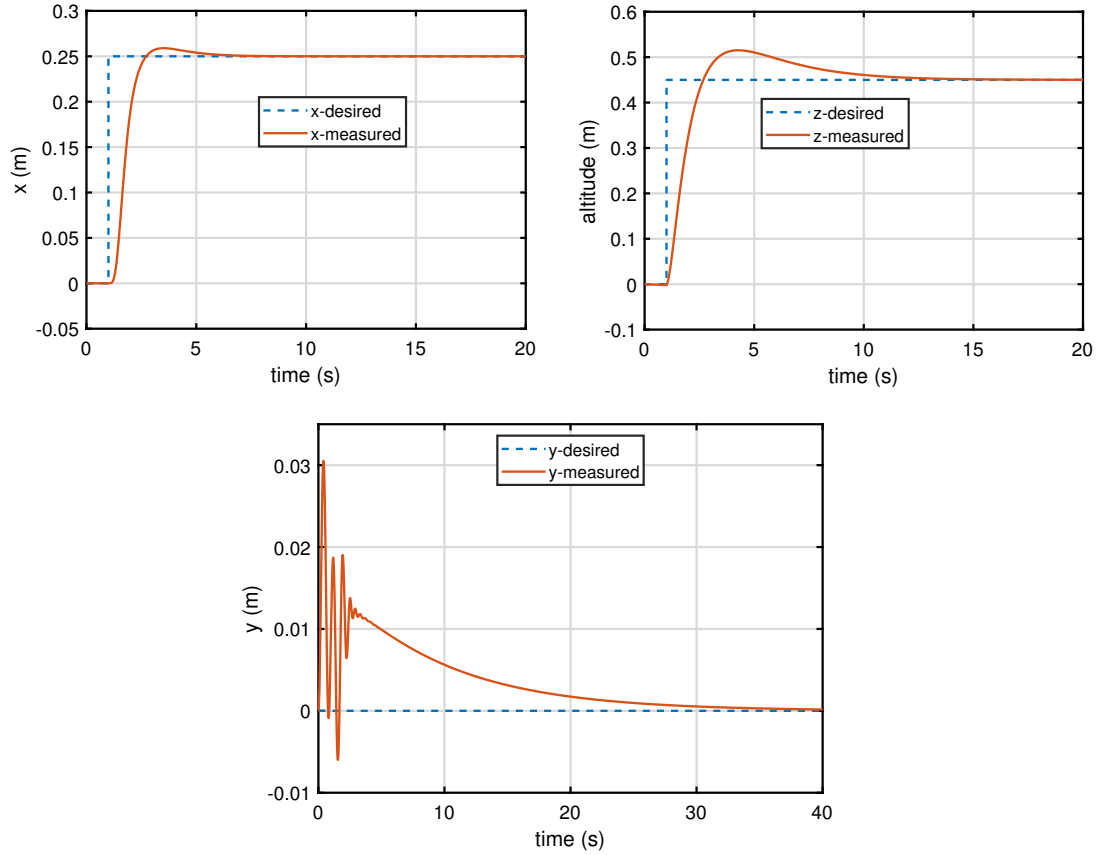
	$(V_x, V_{xd}) \rightarrow \phi_d$	$(V_y, V_{yd}) \rightarrow \theta_d$
$k_P$	1.67	4.39
$k_I$	0	0.52
$k_D$	4.25	0

Table 3.5: PID gains for position loop

	$(x, x_d) \rightarrow V_{xd}$	$(y, y_d) \rightarrow V_{yd}$	$(z, z_d) \rightarrow u_z$
$k_P$	1.652	1.260	4.59
$k_I$	0	0	1.23
$k_D$	0	0	4.25

### 3.5 Simulation Results

A simulation model was built in Matlab/Simulink using the full nonlinear dynamics of the tricopter described in Section 3.2, and following the scheme in Figure 3.7. The maximum speed of the Emax2207-eco 1700KV motors with 6045 propellers on 3S (11.1 Volts) was obtained from experimental data as  $\omega_{max} \approx 1639\text{rad/s}$ . In order to ensure that the control inputs are feasible with respect to the physical constraints of the motor, the following control limits were set:  $u_z \in (-2k_t\omega_{max}^2, 0)$ ,  $u_\phi \in (-l_1k_t\omega_{max}^2, l_1k_t\omega_{max}^2)$ ,  $u_\theta \in (-l_0k_t\omega_{max}^2, 2l_2k_t\omega_{max}^2)$  and  $u_\psi \in (-2k_d\omega_{max}^2, l_0k_t\omega_{max}^2)$ . It is important to mention that these limits were not considered in the controller design. The simulation was run for 40 seconds with the tricopter commanded to move 0.25m in the  $x$  direction and hover at a height of 0.45m.

Figure 3.8:  $x, y$  and  $z$  (altitude) position of tricopter in inertial coordinates.

The results show that the designed controller completely stabilized the tricopter. From Figure 3.8, the tricopter settles at the desired position in the  $x$ -direction after about 5 seconds and reaches the desired altitude in about 10 seconds. There are some oscillations of low magnitude in the measured lateral position but as time tends to infinity, the measured  $y$  position is regulated to zero by the controller. Figure 3.9 shows that the attitude is stabilized within 6 seconds. The initial oscillatory behaviour may be due to the nonlinearities in the plant since the control is based on a linear model, but these oscillations settle within a short period of 3 seconds. The demanded pitch angle is large (over  $45^\circ$ ) initially but this may be due to simulation errors when modelling the tricopter system. Another important factor to consider is that unlike the quadcopter that is inherently stable due to even number of rotors and counter-rotating propellers, the tricopter is inherently unstable. This may account for the initially large demanded pitch angle on take-off.

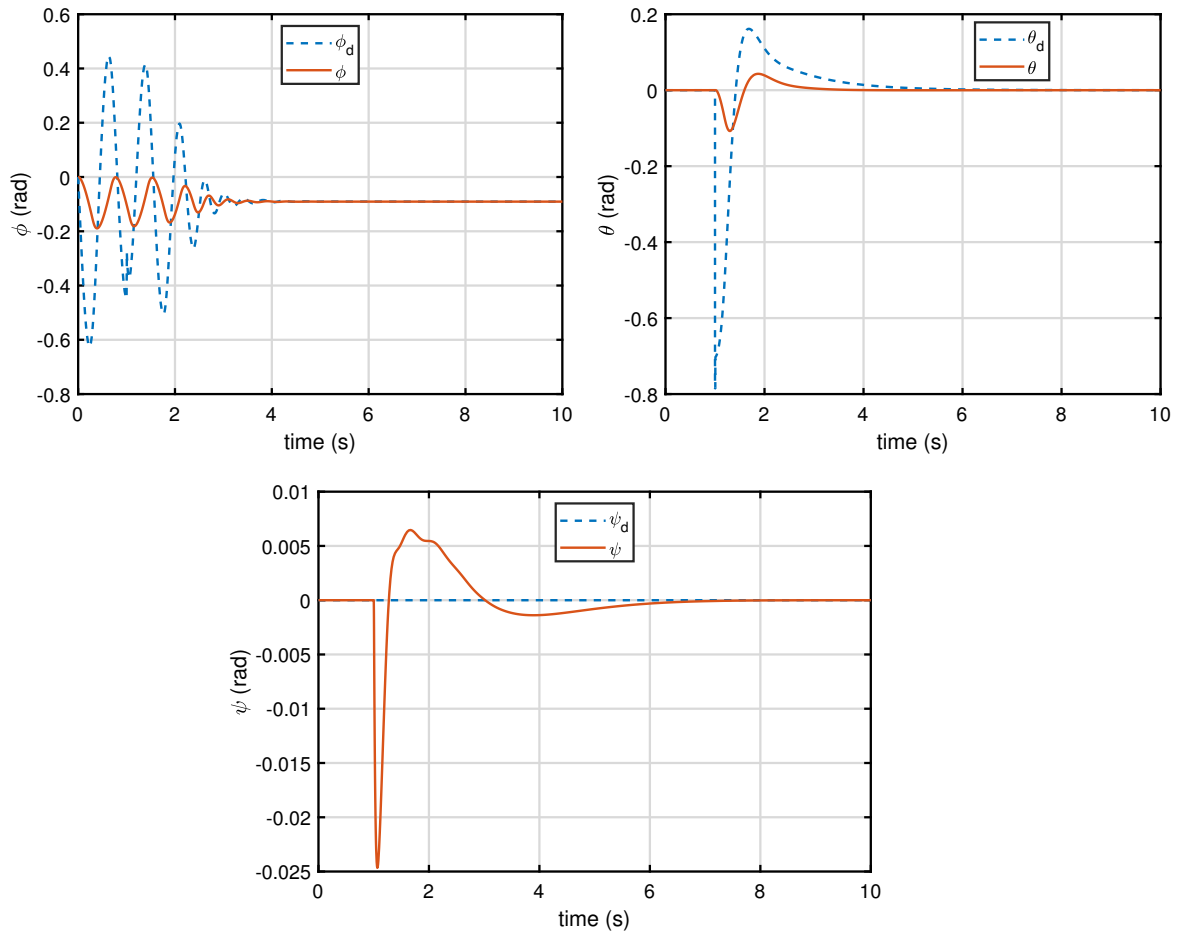


Figure 3.9: Attitude of tricopter.

The control torques ( $\tau_x, \tau_y, \tau_z$ ) are small within a range of  $-0.05$  to  $0.05 \text{ kg.m}^2/\text{s}^2$  as observed from Figure 3.10 and so the controller is practicable. The rotor speeds ( $\omega_1, \omega_2, \omega_3$ ) are also within the physical limits of the selected motor. It is worth noting that the speed of rotor 3 is higher than that of rotors 1 and 2 which are close in value. This higher speed of rotor 3 compared to rotors 1 and 2 is because, to stabilize the yaw attitude,  $\alpha \approx 0.27 \text{ rad}$  (non-zero) in steady hover as observed from Figure 3.10.

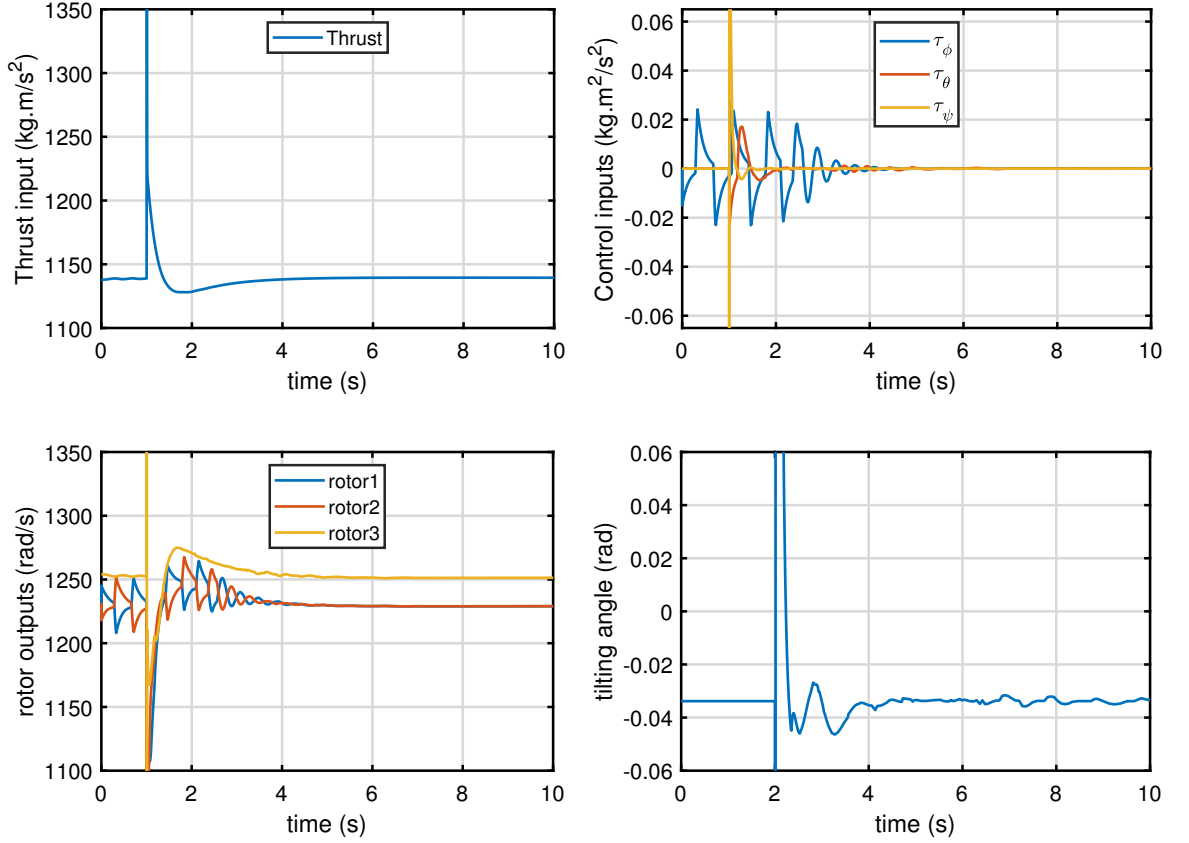


Figure 3.10: Control inputs and rotor speeds of the tricopter.

It then follows from Figure 3.1 that,  $k_t \omega_1^2 \approx k_t \omega_2^2 \approx k_t \omega_3^2 \cos \alpha \approx 2.4 \text{ N}$  in steady hover and for this to be valid, rotor 3 has to spin faster than rotors 1 and 2. It can also be noted that the thrust at hover given as  $F_{\text{hover}} = -m_t g = -7.2422 \text{ kg.m/s}^2$  is also evident from Figure 3.10 which further proves the feasibility of the proposed methods.

## 3.6 Experimental Results

### 3.6.1 Platform Description

The hardware setup of the proposed single-tilt tricopter UAV is depicted in Figure 3.11 and was developed at the Control, Dynamics and Robotics laboratory at the University of Manchester. It weighs 0.739kg, has a triangular structure with three identical arms of length  $l_0$ , with a fixed pitch propeller driven by a Brushless DC motor mounted at the end of each arm.

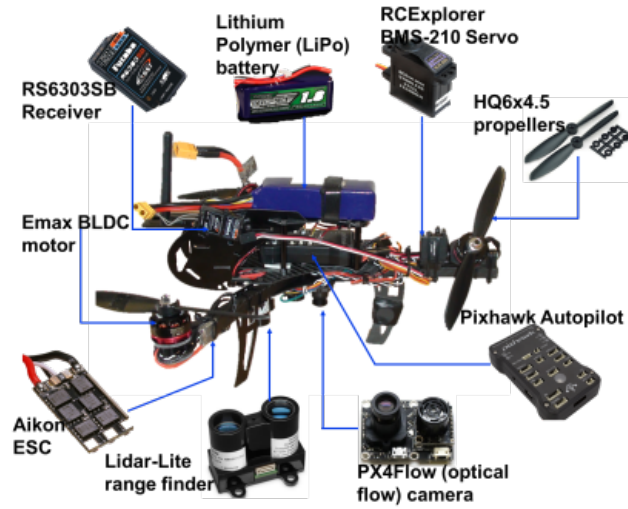


Figure 3.11: Side view of the single-tilt tricopter assembly.

The tilting mechanism of the tail rotor which controls the Yaw motion is a servo-motor to which the propeller-motor assembly is attached. The servo-motor tilts the propeller-motor assembly through  $\alpha \in (-\frac{\pi}{2}, \frac{\pi}{2})$  to generate a lateral component of the generated thrust, thereby generating a yaw torque. See Appendices A and B.1 for a full specification of the single-tilt tricopter.

### 3.6.2 Hover flight test

From trial tests performed, the proposed method is able to stabilize the tricopter's attitude around hover as shown in Figure 3.12 although there are some peaks in roll and pitch, which is expected from the simulations. But this may be due to uncertainties in the plant which have not been considered by the linear model and noisy measurements from the optical flow sensor. Moreover, the PID loops were tuned independently not



considering the interactions and coupling which exist between the loops of the UAV, being that it is a multi-variable system. Also, even though the PWM commands (signals sent to the individual rotors) show some oscillatory behaviour, they are not very noisy and are within configured values of 1000 to 2000 $\mu s$ . This implies a low probability of saturations occurring which is important for good performance. The oscillatory behaviour may be due to inconsistencies in altitude measurements from the range finder since the test was done indoors. Also, the servo angle is tilted (away from the center) on take-off as evident from the PWM value of 1478 $\mu s$  since the standard servo center is approximately 1520 $\mu s$ . The tilt angle changes as necessary at different points during flight in order to stabilize the heading.

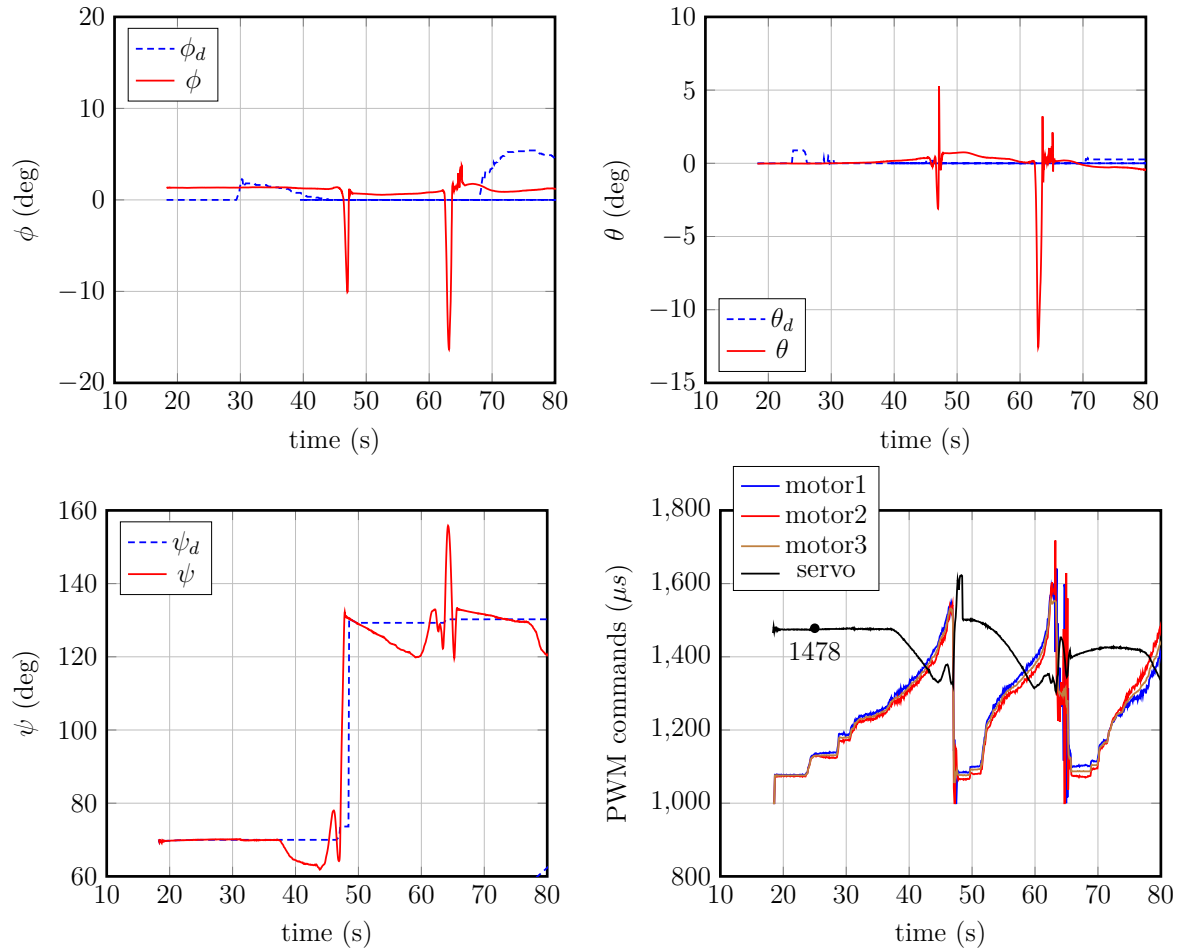


Figure 3.12: Attitude and PWM commands from test.

### 3.7 Summary

In this chapter, a novel tricopter UAV with a single tilt rotor has been developed. The mathematical model has been presented and intuitive methods to obtain the model parameters have been proposed. A control allocation scheme for obtaining motor speeds by inversion of a mixer matrix has also been proposed. These were then used to show how the loops can be closed independently and sequentially using simple PIDs. The proposed methodology aids with an intuitive design which can be tuned easily on practical hardware. The proposed control scheme has been implemented on a simulation model using parameters obtained from the tricopter platform, and some trials have been done on the developed platform for hover control. Although some areas can be improved, the test results are acceptable and provide good grounds for further research into this problem.

# Chapter 4

## Development and control of Multirotor-tilting tricopter with direct actuation for position control

All the material in this chapter was submitted for publication in [79].

### 4.1 Introduction

In this chapter, the hardware for a tricopter vehicle with three independently tilting rotors is developed so as to have a fully actuated vehicle made possible by use of a custom tilt mechanism for the airframe. A quaternion feedback controller is then proposed, which uses a model with attitude represented in unit quaternions to avoid the gimbal lock problem of Euler angles. Experiments to obtain the model parameters from experimental data are developed. The PD-type quaternion feedback controller is used to stabilize the attitude of the tricopter and the performance is tested in simulations. A linear MPC controller for the system is then designed but instead of linearising the system using classic methods based on operating points, the model is put in a Linear Parameter Varying (LPV) form such that the nonlinearities of the tricopter are captured in the model. Linearising around an operating point leads to less precision when the system moves further away from the operating point. Next, the ability of the airframe to translate in the lateral and longitudinal directions without changing its attitude is demonstrated by feedback linearizing the nonlinear translational equations

of the tricopter, to produce the control forces in the  $x$  and  $y$  direction which are directly applied to the tricopter UAV. While this is straight-forward in simulation, it is noted that implementing this in reality is limited by the available radio transmitters which support only 4 degrees of freedom. And to solve this problem, direct actuation is proposed to control the longitudinal and lateral motions via thrust vectoring of the rotors by tilting them in certain directions based on commands from mapped transmitter knobs. Unlike [21] where a specific controller has been proposed, here the position is controlled by implementing direct actuation in the control allocation algorithm via the PX4 flight stack mixing system. The feasibility and applicability of the control allocation is tested indoors on the developed platform test-bed.

The key contributions of this chapter to the state-of-the-art are as follows:

- Design and construction of servo tilt-mechanism which extends the tilt angle range of the rotors in a multi-tilt tricopter.
- Demonstration of the capability of the airframe to translate in the lateral and longitudinal directions without changing its attitude by feedback linearization of the nonlinear translational equations of the tricopter to obtain control forces in  $x$  and  $y$  axes which are directly applied to the tricopter UAV.
- Noting that there is a limitation to implementing full 6 DOF control using regular Radio Transmitters which support only 4 DOF, direct actuation is proposed to control the lateral and longitudinal motions via thrust vectoring of the rotors by tilting them in certain directions based on commands from mapped transmitter knobs. This is implemented in the control allocation algorithm of the PX4 flight stack mixing system, and experiments on the platform test-bed are used to validate the method.

The other technical contributions of this chapter are summarized as follows:

- Different from existing works, a quaternion feedback (QFB) controller which uses a model with attitude represented in unit quaternions is proposed. This avoids the gimbal lock problem of Euler angles.
- Development of identification platform and experiment to obtain model parameters from experimental data.

- Design of linear Model Predictive Controller for the system using a Quasi Linear Parameter Varying (Quasi-LPV) model of the tricopter which captures the nonlinearities of the tricopter, rather than classic linearisation methods based on operating points which leads to less precision when the system moves further away from the operating point. The advantage of the LPV method is that we are able to use a linear control method on the nonlinear system.

The rest of this chapter is organised as follows: The nonlinear dynamic model of the multi-tilt tricopter UAV is described in Section 4.2. The proposed quaternion feedback control scheme is given in Section 4.4 and the LPV-MPC scheme is detailed in Section 4.5. Section 4.6 is devoted to the presentation of the simulation results and comparisons between the control schemes. The full 6-DOF control of the tricopter is demonstrated in Section 4.7. Section 4.8 describes the experimental aspects and a summary is given in Section 4.9.

## 4.2 Modelling of the multirotor-tilting tricopter

In this section, the model for the proposed tricopter hardware is developed. The forces and moments which act on the proposed airframe are derived first, and then the kinematics and rigid-body dynamics are developed. Note that a similar model had been proposed in [20] and [80].

### 4.2.1 Tricopter Forces and Moments

The proposed airframe is given in Figure 4.1 with 3 rotors each mounted on a servo and whose propellers rotate in the clockwise direction. As shown in Figure 4.1,  $(X^e, Y^e, Z^e)$  denotes the earth coordinate system which is assumed to be inertial,  $(X^b, Y^b, Z^b)$  denotes the body coordinate system with its origin fixed to the center of mass  $\mathbf{G}$  of the vehicle and  $(X^{l^1}, Y^{l^2}, Z^{l^3})$  denotes the local coordinate system with its origin at  $(O_1, O_2, O_3)$  which are the application points of the forces  $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3$  respectively. Also, Figure 4.2 shows the top view of the same coordinate systems. The reader should note that diagram in Figure 3.2 applies to the multi-tilt tricopter as well. The individual uni-directional forces and drag torques produced by each rotor

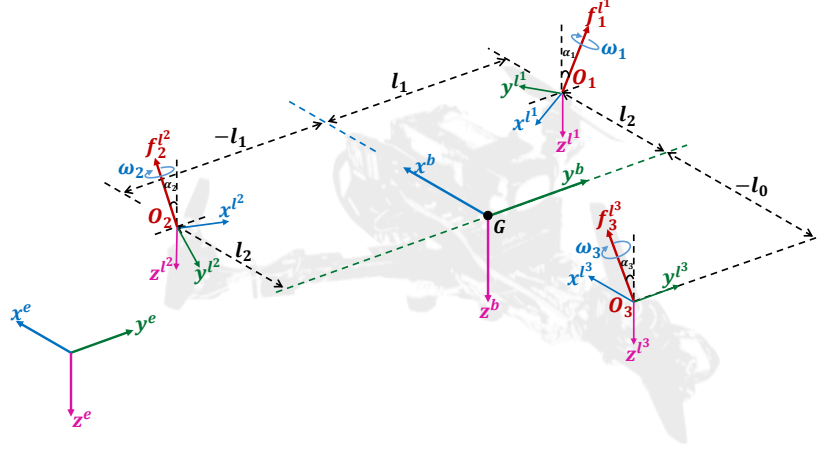


Figure 4.1: Forces and torques acting on tricopter and coordinate systems

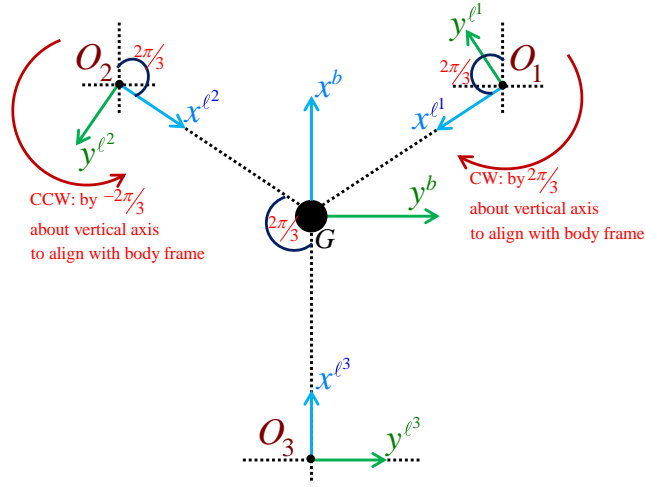


Figure 4.2: Top view of the tricopter showing all frames where CW and CCW mean clockwise and counter clockwise respectively.

are assumed to be proportional to the square of the angular speeds  $\omega_i$  [76] since the propeller is directly coupled with the DC motor so that

$$f_i = k_t \omega_i^2 \text{ and } \tau_i = k_d \omega_i^2 \quad \forall i \in \{1, 2, 3\} \quad (3.1 \text{ revisited})$$

where  $f_i$  and  $\tau_i$  denote the forces and drag torques respectively,  $k_t$  is the thrust constant and  $k_d$  is the drag-torque constant. The 3D force produced by the  $i^{\text{th}}$  rotor from Figure 4.1 is

$$\mathbf{f}_i^{l_i} = \begin{bmatrix} 0 \\ -k_t \omega_i^2 \sin \alpha_i \\ -k_t \omega_i^2 \cos \alpha_i \end{bmatrix}, \quad \alpha_i \in [-\pi/2, \pi/2] \quad \forall i \in \{1, 2, 3\}, \quad (4.1)$$

and this is given relative to the local frame with  $\alpha_i$  as the angle of tilt about each local  $x$  axis. In order to align the rotor forces  $\mathbf{f}_i^{l_i} \quad \forall i \in \{1, 2\}$  with the body frame,

$\mathbf{f}_1^{l_1}$  is rotated clockwise through  $\psi = 2\pi/3$  while  $\mathbf{f}_2^{l_2}$  is rotated counter-clockwise through  $\psi = -2\pi/3$  (as depicted in Figure 4.2) corresponding to  $\mathbf{q}_1 = \left[\frac{1}{2} \ 0 \ 0 \ \frac{\sqrt{3}}{2}\right]^T$  and  $\mathbf{q}_2 = \left[\frac{1}{2} \ 0 \ 0 \ -\frac{\sqrt{3}}{2}\right]^T$  respectively using (2.37), to yield the following rotation matrices,

$$\mathbf{R}_{l_1}^b = \begin{bmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{R}_{l_2}^b = \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

so that  $\mathbf{f}_i^b = \mathbf{R}_{l_i}^b \mathbf{f}_i^{l_i}$  for  $i \in \{1, 2\}$ . Note that  $\mathbf{f}_3^b = \mathbf{f}_3^{l_3}$ . The total force from the three rotors is then computed as

$$\begin{aligned} \mathbf{F}_m^b &= \sum_{i=1}^3 \mathbf{f}_i^b \\ &= \begin{bmatrix} -\frac{\sqrt{3}}{2}f_1 \sin \alpha_1 + \frac{\sqrt{3}}{2}f_2 \sin \alpha_2 \\ \frac{1}{2}f_1 \sin \alpha_1 + \frac{1}{2}f_2 \sin \alpha_2 - f_3 \sin \alpha_3 \\ -f_1 \cos \alpha_1 - f_2 \cos \alpha_2 - f_3 \cos \alpha_3 \end{bmatrix}. \end{aligned} \quad (4.2)$$

Let  $\mathbf{G}_{o_i} = [G_{o_{ix}} \ G_{o_{iy}} \ G_{o_{iz}}]^T$  be the vector of the  $i^{\text{th}}$  rotor's distance from the center of mass  $\mathbf{G}$ , and  $(O_1, O_2, O_3)$  be the application points of  $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3$  respectively. It follows from Figure 4.1 that  $\mathbf{G}_{o_1} = [l_2, \ l_1, \ 0]^T$ ,  $\mathbf{G}_{o_2} = [l_2, \ -l_1, \ 0]^T$  and  $\mathbf{G}_{o_3} = [-l_0, \ 0, \ 0]^T$  where  $l_0$  is the length of each rotor arm,  $l_1 = \frac{\sqrt{3}}{2}l_0$  and  $l_2 = \frac{1}{2}l_0$ . Then, the total 3D moments produced in the body-fixed frame is

$$\begin{aligned} \mathbf{T}_r^b &= \sum_{i=1}^3 (\mathbf{G}_{o_i} \times \mathbf{f}_i^b) \\ &= \begin{bmatrix} \frac{\sqrt{3}}{2}l_0f_2 \cos \alpha_2 - \frac{\sqrt{3}}{2}l_0f_1 \cos \alpha_1 \\ \frac{1}{2}l_0f_1 \cos \alpha_1 + \frac{1}{2}l_0f_2 \cos \alpha_2 - l_0f_3 \cos \alpha_3 \\ l_0f_1 \sin \alpha_1 + l_0f_2 \sin \alpha_2 + l_0f_3 \sin \alpha_3 \end{bmatrix}. \end{aligned} \quad (4.3)$$

Like the rotor force, the drag torque acts in an opposite direction to the  $i^{\text{th}}$  rotor's spin direction and is given as

$$\boldsymbol{\tau}_{d,i}^l = \begin{bmatrix} 0 \\ -k_d \omega_i^2 \sin \alpha_i \\ -k_d \omega_i^2 \cos \alpha_i \end{bmatrix} \quad \forall i \in \{1, 2, 3\}, \quad (4.4)$$

given relative to the local frame. The total drag-torque produced in the body frame is thus

$$\begin{aligned}\mathbf{T}_d^b &= \sum_{i=1}^3 (\mathbf{R}_{l_i}^b \boldsymbol{\tau}_{d,i}^{l_i}) \\ &= \begin{bmatrix} -\frac{\sqrt{3}}{2}\tau_1 \sin \alpha_1 + \frac{\sqrt{3}}{2}\tau_2 \sin \alpha_2 \\ \frac{1}{2}\tau_1 \sin \alpha_1 + \frac{1}{2}\tau_2 \sin \alpha_2 - \tau_3 \sin \alpha_3 \\ -\tau_1 \cos \alpha_1 - \tau_2 \cos \alpha_2 - \tau_3 \cos \alpha_3 \end{bmatrix},\end{aligned}\quad (4.5)$$

so that the total torque acting on the tricopter is

$$\mathbf{T}_m^b = \mathbf{T}_r^b + \mathbf{T}_d^b \quad (4.6)$$

Finally, combining (4.2) and (4.6), the relationship between the control inputs and actuator outputs may be represented as

$$\mathbf{U} = \mathbf{M}\boldsymbol{\Omega} \quad (4.7)$$

where

$$\mathbf{M} = \begin{bmatrix} -\frac{\sqrt{3}}{2}k_t & \frac{\sqrt{3}}{2}k_t & 0 & 0 & 0 & 0 \\ \frac{1}{2}k_t & \frac{1}{2}k_t & -k_t & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_t & -k_t & -k_t \\ -\frac{\sqrt{3}}{2}k_d & \frac{\sqrt{3}}{2}k_d & 0 & -\frac{\sqrt{3}}{2}k_t l_0 & \frac{\sqrt{3}}{2}k_t l_0 & 0 \\ \frac{1}{2}k_d & \frac{1}{2}k_d & -k_d & \frac{1}{2}k_t l_0 & \frac{1}{2}k_t l_0 & -k_t l_0 \\ k_t l_0 & k_t l_0 & k_t l_0 & -k_d & -k_d & -k_d \end{bmatrix}, \quad \boldsymbol{\Omega} = \begin{bmatrix} \omega_1^2 \sin \alpha_1 \\ \omega_2^2 \sin \alpha_2 \\ \omega_3^2 \sin \alpha_3 \\ \omega_1^2 \cos \alpha_1 \\ \omega_2^2 \cos \alpha_2 \\ \omega_3^2 \cos \alpha_3 \end{bmatrix}$$

and  $\mathbf{U} = [\mathbf{F}_m^b \quad \mathbf{T}_m^b]^\top$ .

**Remark 4.1.** Considering (4.7), the matrix  $\mathbf{M}$  represents the relationship between the control inputs  $\mathbf{U}$  and the actuator control outputs  $\boldsymbol{\Omega}$  which are the angular speeds and tilt angles. As there are 6 available control inputs as evident from the rows of  $\mathbf{M}$ , the multi-tilt tricopter is fully actuated because all of its 6-DOF can be independently controlled. The first 2 rows of  $\mathbf{M}$  represent the translational forces (lateral and longitudinal) making it feasible to compute the forces needed to directly control the lateral and longitudinal motions of the tricopter without varying the roll, pitch or yaw angles as is the case with other classic multicopters where the longitudinal and lateral motions cannot be independently controlled.



To check if the matrix  $\mathbf{M}$  is invertible, its row echelon form  $\mathbf{M}_{\text{REC}}$  was obtained via row reduction (see Appendix D for details) and is given by,

$$\mathbf{M}_{\text{REC}} = \begin{bmatrix} -\frac{\sqrt{3}}{2}k_t & \frac{\sqrt{3}}{2}k_t & 0 & 0 & 0 & 0 \\ 0 & k_t & -k_t & 0 & 0 & 0 \\ 0 & 0 & 3k_tl_0 & -k_d & -k_d & -k_d \\ 0 & 0 & 0 & -\frac{\sqrt{3}}{2}k_tl_0 & \frac{\sqrt{3}}{2}k_tl_0 & 0 \\ 0 & 0 & 0 & 0 & k_tl_0 & -k_tl_0 \\ 0 & 0 & 0 & 0 & 0 & -3k_t \end{bmatrix}.$$

It is evident that  $\mathbf{M}_{\text{REC}}$  is full rank if  $k_t, l_0 > 0$  and since  $k_t$  and  $l_0$  are positive constants,  $\mathbf{M}$  is invertible according to definitions 2.2 and 2.3.

### 4.2.2 Dynamic model

Similar to the single-tilt case, the multi-tilt tricopter is considered to be a rigid-body with mass  $m_t$  and the total force acting on the tricopter body  $\mathbf{F}_t^b$  is the sum of the rotor force  $\mathbf{F}_m^b$  and the force of gravity  $\mathbf{F}_g^e = [0 \ 0 \ m_t g]^\top$  where  $g$  is the gravity constant. By relating (2.21) and (2.38), the attitude dynamics is given in terms of quaternion as,

$$\begin{aligned} \dot{\mathbf{q}} &= \frac{1}{2} \mathbf{Q}(\mathbf{q}) \begin{bmatrix} 0 \\ \boldsymbol{\omega}^b \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1}{2} \mathbf{q}_v \boldsymbol{\omega}^b \\ \frac{1}{2} (\mathbf{S}(\mathbf{q}_v) + q_0 \mathbf{I}_3) \boldsymbol{\omega}^b \end{bmatrix}, \end{aligned} \quad (4.8)$$

where  $\boldsymbol{\omega}^b$  is the angular velocity. The other dynamic equations of the multi-tilt tricopter are identical to those of the single-tilt case as given in subsection 3.2.2.

## 4.3 Parameter Identification

In this section, a series of experiments are developed and carried out to identify the model parameters of the proposed platform. In Section 4.3.1, the moments of inertia for the tricopter's three main axes are identified while in Section 4.3.2 the constants of the DC motor are identified.

### 4.3.1 Moment of Inertia

An experiment based on the trifilar pendulum method was developed and used to obtain the moments of inertia across the tricopter's axes. Rather than using a CAD model for the multi-tilt tricopter as done in the single-tilt case, this method was used because it was relatively simple and easier to setup but despite its simplicity, this method allows for very high accuracy [81]. Furthermore, the multi-tilt tricopter has a more complex configuration and developing a precise CAD model capable of providing useful estimates of the moments of inertia would require a considerable amount of time. These two reasons motivated the choice of using the trifilar method. The experimental platform as depicted in Figures 4.3 and 4.4 consists of a plastic hoop of mass  $m_h = 0.105\text{kg}$  suspended by three parallel steel wire ropes of length  $L = 1\text{m}$  such that the perpendicular distance of each wire to the vertical axis through the center of gravity is the radius of the hoop  $R = 0.25\text{m}$ , and the angle between the wires is  $120^\circ$ .

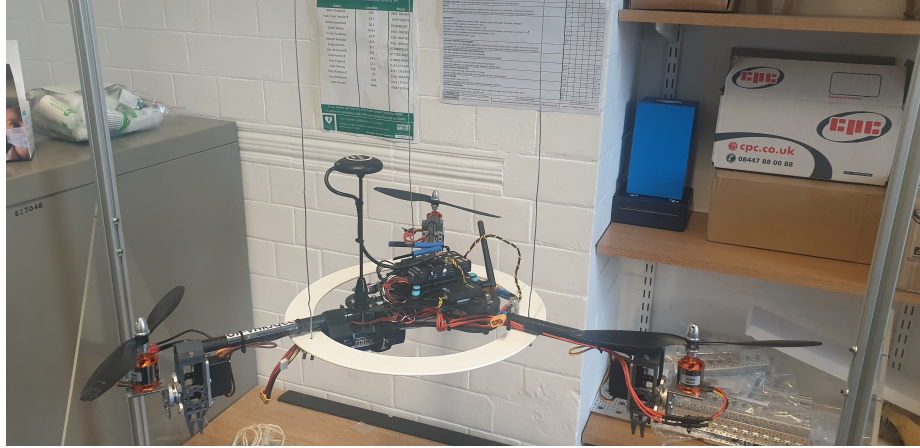


Figure 4.3: Trifilar experiment with tricopter set for measuring  $J_z$

The body whose inertia is to be measured is placed on the hoop with the axis along which the inertia needs to be found aligned parallel to the wires that is, parallel to the vertical axis. The moment of inertia about the vertical axis can then be obtained from the relation [82]

$$J_z = \frac{mgR^2T^2}{4\pi^2L}, \quad (4.9)$$

where  $m$  is the mass of the body whose inertia is to be measured and  $T$  is the period of 1 oscillation. Noting that the hoop of the platform also has inertia, the inertia of the hoop is separated from that of the tricopter body, so that the moment of inertia

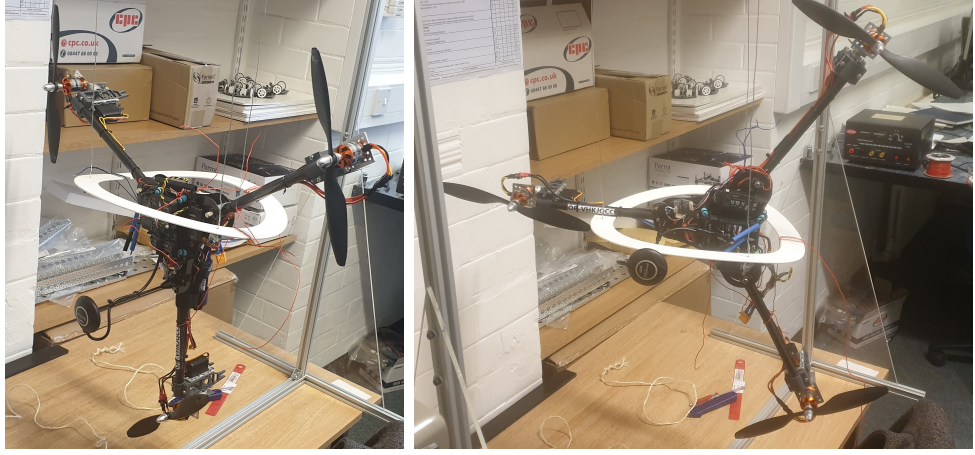


Figure 4.4: Trifilar experiment with tricopter set for measuring  $J_x$  and  $J_y$

of the tricopter only, along the vertical axis becomes

$$J_{tz} = \frac{gR^2T^2}{4\pi^2L}(m_h + m_t) - J_{hz}, \quad (4.10)$$

where  $J_{hz}$  is the inertia of the unloaded hoop obtained using (4.9). This was done to reduce error in measurements thereby improving the accuracy of the experiment. The tricopter's mass including the battery was measured using a weighing scale and obtained as  $m_t = 1.448\text{kg}$ . Let us start with the moment of inertia around the  $z$  axis. The tricopter was placed on the hoop with the  $z$  axis aligned parallel to the wires that is, parallel to the vertical axis, and then a small disturbance was applied to make the body oscillate. The time taken for 10 oscillations was then recorded using a stop-watch. Similarly, the  $x$  and  $y$  axes of the tricopter were each aligned parallel to the wires (refer to Figure 4.4) and in each case, the time taken for 10 oscillations around the vertical axis was recorded. Five trials were done and the average taken to reduce random errors in the measurement so as to improve accuracy and this was used to obtain the period for 1 oscillation. Table 4.1 shows the obtained measurements and the moments of inertia were then computed as  $J_x = 0.1035 \text{ kg.m}^2$ ,  $J_y = 0.1031 \text{ kg.m}^2$  and  $J_z = 0.1709 \text{ kg.m}^2$  so that  $\mathbf{J} = \text{diag}\{J_x, J_y, J_z\}$ .

**Remark 4.2.** It should be noted that no comparison has been made to the CAD method of estimating moments of inertia used in chapter 3. This is because the CAD method was used on the single-tilt tricopter which is a different platform to the multi-tilt tricopter. Also, an error analysis of the trifilar method has not been performed here. Being an experimental method, there will be errors introduced into the measurements

from different parts of the setup. The interested reader should therefore see [81, 83] for an analysis of the errors in a trifilar inertia experiment. Also, for a general review and comparison of methods used in estimating moments of inertia see [84].

### 4.3.2 Thrust and Torque Constants

An experiment consisting of an RCBenchmark thrust stand and dynamometer fitted to a wooden board and mounted on a bench was setup as in Figure 3.5 just like in subsection 3.3.2 for measuring the thrust, torque and speed. The constants are then obtained by plotting the thrust and drag torque against the square of the speed respectively, so that  $k_t$  and  $k_d$  are gradients of the best fitting line through the data points, constructed via least squares. The results are shown in Figure 4.5 as  $k_t = 1.084 \pm 0.1 \times 10^{-5} \text{ N}/(\text{rad/s})^2$  and  $k_d = 1.726 \pm 0.001 \times 10^{-7} \text{ N.m}/(\text{rad/s})^2$  when using FlashHobby D2836 880KV motors with  $10 \times 4.5$  propellers (10 inches long with 4.5 degrees pitch angle) on 4S (14.8 Volts).

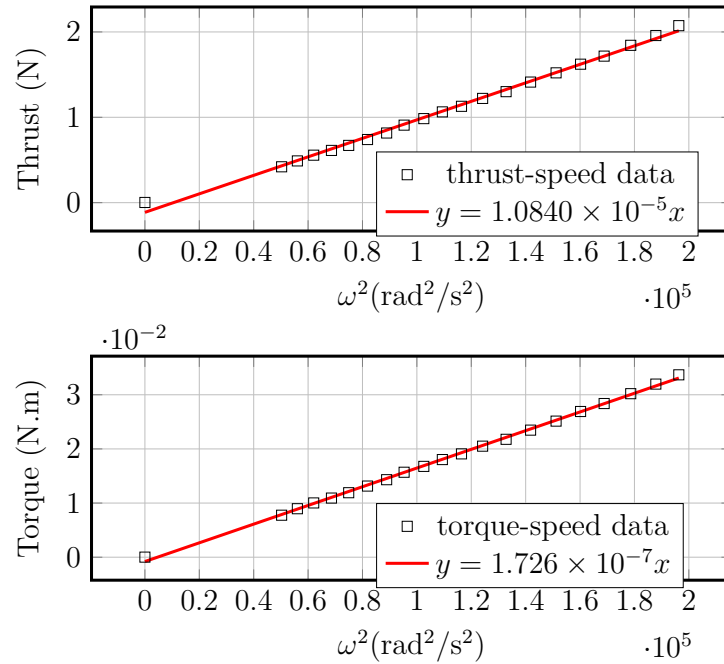


Figure 4.5: Thrust and drag-torque constants data with 93.82% fit for  $k_t$  and 96.78% fit for  $k_d$ .

Table 4.1: Trifilar experiment: Time for 10 oscillations. Ax means axis hence  $t_x$  is the measured time for 10 oscillations around x axis,  $h_z$  is the measured time for 10 oscillations of the empty hoop around the z-axis. All measurements are in seconds.

Ax	Five Trials					Mean	$T$
$t_x$	21.65	21.43	21.37	21.42	21.32	21.438	2.144
$t_y$	21.58	21.57	21.16	21.32	21.37	21.400	2.140
$t_z$	27.79	27.70	27.50	27.49	27.29	27.554	2.755
$h_z$	17.43	17.02	17.23	17.11	17.02	17.162	1.716

## 4.4 Attitude control using Quaternion feedback control (QFB) technique

In this section, a nonlinear PD-type feedback controller based on quaternions similar to the work of [85], is proposed to stabilize the attitude of the tricopter as depicted in Figure 4.6.

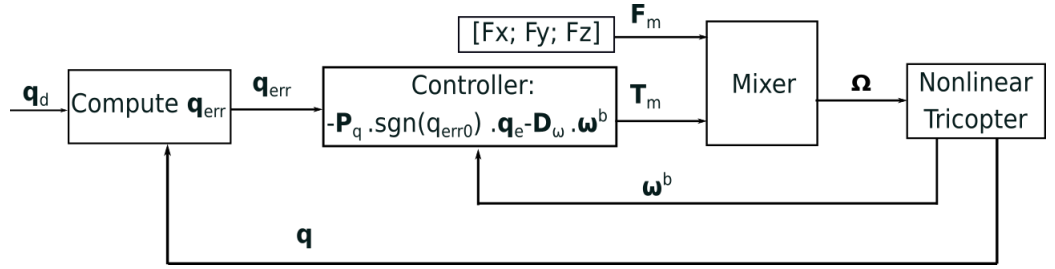


Figure 4.6: Quaternion feedback control scheme.

### 4.4.1 QFB control scheme

The scheme in Figure 4.6 consists of an inner angular rate controller  $D_w$  and an outer quaternion attitude controller  $P_q$ . The *Nonlinear Tricopter* block represents the model of the tricopter, the *mixer* block represents the control allocation which is the computation of the actuator outputs from the control inputs achieved using (4.7) and the *Compute  $q_{err}$*  block represents the computation of the quaternion error vector. The *Controller* block represents the QFB controller which is mathematically denoted by,

$$\tau = -P_q \text{sgn}(q_{err0}) q_e - D_w \omega^b \quad (4.11)$$

where  $P_q \in \mathbb{R}^{3 \times 3}$ ,  $D_w \in \mathbb{R}^{3 \times 3}$  are diagonal matrices,  $\text{sgn}$  represents the signum function,  $q_{err0}$  and  $q_e \in \mathbb{R}^{3 \times 1}$  are the scalar and vector components respectively, of the error

quaternion (discussed later) in body frame. As noted in [86], selecting  $\mathbf{P}_q = 2\omega_n^2 \mathbf{I}_3$  and  $\mathbf{D}_w = 2\zeta\omega_n \mathbf{I}_3$  yields stability of the closed loop and this is used as a starting point in tuning the controllers. The natural frequency  $\omega_n$  affects the speed of the response and  $\zeta$  is the damping ratio. Designing for settling time  $t_s$  of 4s and using the relation  $t_s \approx \frac{4}{\zeta\omega_n}$  [78], we obtain a requirement of  $\omega_n = 1.5\text{rad/s}$  for  $\zeta = 0.65\text{s}$ .

#### 4.4.2 Quaternion error

The quaternion error is the relative rotation between a desired attitude  $\mathbf{q}_d$  and actual attitude  $\mathbf{q}$ . A commonly used formulation presented in [4] and [86] is utilised to form a set of error angles as  $\mathbf{q}_{err} = \mathbf{q}_d^{-1} \otimes \mathbf{q} = q_{err0} + \mathbf{q}_e$ . This implies,

$$\begin{aligned} \mathbf{q}_{err} &= \mathbf{q}_d^* \otimes \mathbf{q}, \\ &= \mathbf{Q}(\mathbf{q}_d)^\top \mathbf{q}, \\ &= \begin{bmatrix} q_{0_d} & q_{1_d} & q_{2_d} & q_{3_d} \\ -q_{1_d} & q_{0_d} & q_{3_d} & q_{2_d} \\ -q_{2_d} & -q_{3_d} & q_{0_d} & q_{1_d} \\ -q_{3_d} & q_{2_d} & -q_{1_d} & q_{0_d} \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \end{aligned} \quad (4.12)$$

so that,

$$q_{err0} = q_{0_d}q_0 + q_{1_d}q_1 + q_{2_d}q_2 + q_{3_d}q_3. \quad (4.13)$$

It is worth noting that all unit quaternions possess the double cover property over the  $SO(3)$  group of rotations [87] and if the reference is demanding a rotation greater than  $\pi$  radians, it is ensured that the vehicle is always commanded to rotate in the shortest direction by setting the sign of  $\mathbf{P}_q$  in (4.11) to be the same as that of  $q_{err0}$  hence the reason for the term  $\text{sgn}(q_{err0})$ . However, arbitrarily small disturbances might trap the system at  $\pi$  radians as noted in [88].

It is also important to mention that from a practical perspective, switching controllers have several issues which ultimately determine the applicability of the control technique. One of such issues is robustness as noted in [89]. While the concept of robustness with respect to parameter variations is well-defined for linear time invariant systems, it is more difficult to quantify for switched linear systems. The presence of the switching element in the loop causes transients which have an effect on controller realisation.

## 4.5 Attitude control using MPC

In this section, a linear MPC based on a Linear Parameter Varying (LPV) model is designed to stabilise the attitude of the tricopter. The scheme is given in Figure 4.7 and it can be seen that the MPC has several blocks. For linear control, we normally have to first obtain the linearised equations of the model. The issue is that linearisation makes the control less precise when the system moves away from the operating point. For this reason, the system is instead put into a Linear Parameter Varying (LPV) form which encapsulates the tricopter's nonlinearities. As MPC is more complex, Euler angles are used for simplicity.

### 4.5.1 Quasi-LPV modelling of the tricopter

Since the main concern is with attitude control close to the hovering point, only the rotational equations of the tricopter are considered and an assumption is made that the angles  $\phi$  and  $\theta$  are small so that from (2.41),  $\mathbf{\Gamma} \approx \mathbf{I}_3$ . Consequently,  $\dot{\boldsymbol{\eta}} = \boldsymbol{\omega}^b$  from (3.10) so that  $\ddot{\boldsymbol{\eta}} = \dot{\boldsymbol{\omega}}^b$ . Then using this to relate (3.10) and (3.11) and expanding, we have

$$\begin{aligned}\ddot{\phi} &= \frac{J_y - J_z}{J_x} \dot{\theta} \dot{\psi} + \frac{\tau_1}{J_x}, \\ \ddot{\theta} &= \frac{J_z - J_x}{J_y} \dot{\phi} \dot{\psi} + \frac{\tau_2}{J_y}, \\ \ddot{\psi} &= \frac{J_x - J_y}{J_z} \dot{\phi} \dot{\theta} + \frac{\tau_3}{J_z}.\end{aligned}\tag{4.14}$$

The states, inputs and outputs are chosen respectively as  $\mathbf{x} = [\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi}]^\top$ ,  $\mathbf{u} = [\tau_1 \ \tau_2 \ \tau_3]^\top$  and  $\mathbf{y} = [\phi \ \theta \ \psi]^\top$  so that the system (4.14) can be put in the LPV form

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}(\dot{\phi}, \dot{\theta}) \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t), \\ \mathbf{y} &= \mathbf{C} \mathbf{x}(t),\end{aligned}\tag{4.15}$$

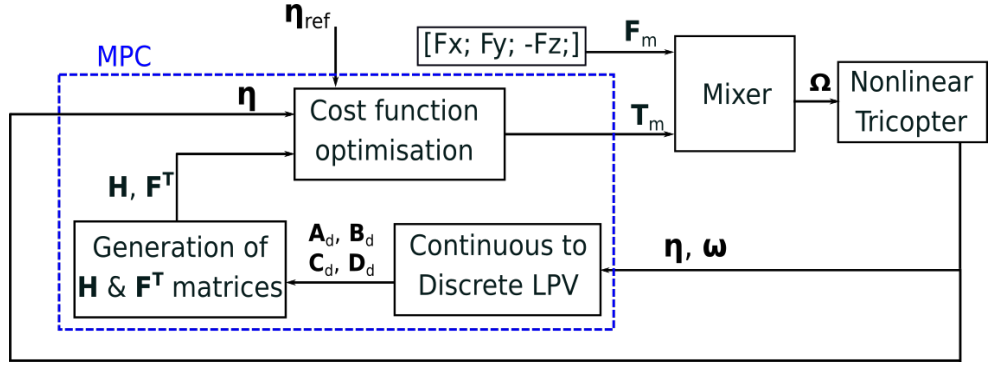


Figure 4.7: LPV-MPC scheme

where

$$\mathbf{A}(\dot{\phi}, \dot{\theta}) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{J_y - J_z}{J_x} \dot{\theta} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{J_z - J_x}{J_y} \dot{\phi} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{J_x - J_y}{2J_z} \dot{\theta} & 0 & \frac{J_x - J_y}{2J_z} \dot{\phi} & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{J_x} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J_z} \end{bmatrix},$$

and

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

As seen from the system matrix  $\mathbf{A}$ , the nonlinearities of the system (4.14) are captured in the LPV form implying that the matrix  $\mathbf{A}$  will be updated at each time step and the MPC will compute a new input at each step.

### 4.5.2 MPC formulation

The MPC problem is formulated as a quadratic optimisation problem of the form

$$J = \min_{\Delta \mathbf{u}_k} \frac{1}{2} \sum_{i=0}^{N-1} \left( \mathbf{e}_{k+i}^\top \mathbf{V} \mathbf{e}_{k+i} + \Delta \mathbf{u}_{k+i}^\top \mathbf{W} \Delta \mathbf{u}_{k+i} \right) + \frac{1}{2} \mathbf{e}_{k+N}^\top \mathbf{Z} \mathbf{e}_{k+N} \quad (4.16)$$

$$\text{s.t. } \mathbf{x}_{k+1} = \mathbf{A}^d(\dot{\phi}, \dot{\theta}) \mathbf{x}_k + \mathbf{B}^d \Delta \mathbf{u}_k, \quad \mathbf{y}_k = \mathbf{C}^d \mathbf{x}_k, \quad (4.17)$$

where  $k$  is the present state and  $k+i$  is  $i$  samples away from the present state,  $\mathbf{x} \in \mathbb{R}^n$  is the state vector,  $\mathbf{e}_k \in \mathbb{R}^m$  is the error vector of the angles which we want to drive to 0,  $\Delta \mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1} \in \mathbb{R}^m$  is the change in inputs,  $\mathbf{V} \in \mathbb{R}^{m \times m}$ ,



$\mathbf{W} \in \mathbb{R}^{m \times m}$  and  $\mathbf{Z} \in \mathbb{R}^{m \times m}$  are positive definite weights,  $\mathbf{A}^d \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B}^d \in \mathbb{R}^{n \times m}$  and  $\mathbf{C}^d \in \mathbb{R}^{m \times n}$  are matrices of the system with  $n$  states and  $m$  inputs and outputs, and  $N$  is the prediction horizon which defines how many future samples of the states are predicted.

**Remark 4.3.** For reference tracking ( $\mathbf{e}_k = 0$ ),  $\mathbf{u}_k \neq 0$  but in the optimisation if  $\mathbf{e}_k = 0$  and we use  $\mathbf{u}_k$  as the input, then the error terms in (4.16) will be 0 but the input terms will not tend to 0. The optimiser will then try to drive  $\mathbf{u}_k$  to 0 resulting in  $\mathbf{e}_k \neq 0$  at steady state [90]. Hence, minimising the change in input  $\Delta \mathbf{u}_k$  is preferred since  $\Delta \mathbf{u}_k = 0$  when  $\mathbf{e}_k = 0$ .

Since the MPC algorithm is in discrete time, the LPV system (4.15) is discretized using the forward Euler method [91] and then to account for the new input  $\Delta \mathbf{u}_k$  in the system (4.15), the input one sample in the past  $\mathbf{u}_{k-1}$  is selected as an extra state to yield the augmented discrete time system as

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_k \end{bmatrix} &= \begin{bmatrix} \mathbf{A}^d & \mathbf{B}^d \\ \mathbf{0}_{m \times n} & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{B}^d \\ \mathbf{I}_m \end{bmatrix} \Delta \mathbf{u}_k, \\ \mathbf{y}_k &= [\mathbf{C}^d \quad \mathbf{0}_3] \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_{k-1} \end{bmatrix}, \end{aligned} \quad (4.18)$$

where  $\mathbf{A}^d(\dot{\phi}, \dot{\theta}) = \mathbf{I}_n + \mathbf{A}(\dot{\phi}, \dot{\theta})\Delta t$ ,  $\mathbf{B}^d = \mathbf{B}\Delta t$  and  $\mathbf{C}^d = \mathbf{C}$  (see Appendix C.1 for discretization process) are the discretized matrices and  $\Delta t$  is the sample time. More compactly, (4.18) is given as

$$\begin{aligned} \tilde{\mathbf{x}}_{k+1} &= \tilde{\mathbf{A}}(\dot{\phi}, \dot{\theta})_{k+1} \tilde{\mathbf{x}}_k + \tilde{\mathbf{B}} \Delta \mathbf{u}_k, \\ \tilde{\mathbf{y}}_k &= \tilde{\mathbf{C}} \tilde{\mathbf{x}}_k, \end{aligned} \quad (4.19)$$

where  $\tilde{\mathbf{A}}(\dot{\phi}, \dot{\theta}) \in \mathbb{R}^{(n+m) \times (n+m)}$ ,  $\tilde{\mathbf{B}} \in \mathbb{R}^{(n+m) \times m}$ ,  $\tilde{\mathbf{C}} \in \mathbb{R}^{m \times (n+m)}$  and  $\tilde{\mathbf{x}}_k \in \mathbb{R}^{(n+m) \times 1}$ . Using the augmented system, the attitude error vector may be defined as  $\mathbf{e}_k = \mathbf{r}_k - \tilde{\mathbf{C}} \tilde{\mathbf{x}}_k$  where  $\mathbf{r}_k$  is the demanded attitude and by substituting this in (4.16) and expanding (see Appendix C.2.1), the cost function becomes

$$\begin{aligned} J = \min_{\Delta \mathbf{u}_k} \sum_{i=1}^{N-1} & \left[ -\mathbf{r}_{k+i}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+i} + \frac{1}{2} \tilde{\mathbf{x}}_{k+i}^\top \tilde{\mathbf{C}}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+i} \right] + \sum_{i=0}^{N-1} \left[ \frac{1}{2} \Delta \mathbf{u}_{k+i}^\top \mathbf{W} \Delta \mathbf{u}_{k+i} \right] \\ & - \frac{1}{2} \mathbf{r}_{k+N}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N} + \frac{1}{2} \tilde{\mathbf{x}}_{k+N}^\top \tilde{\mathbf{C}}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N} + c_1. \end{aligned} \quad (4.20)$$

Note that  $c_1$  represents constant offsets which are ignored since in a minimization problem constants play no role and do not affect the result and from [92], it is standard practice to ignore them. Instead of using the summation operator, we may present (4.20) by stacking the future values of  $\mathbf{r}$ ,  $\Delta \mathbf{u}$  and  $\tilde{\mathbf{x}}$  into long global vectors containing all samples for a given horizon period  $N$  as

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_{k+1} \\ \mathbf{r}_{k+2} \\ \vdots \\ \mathbf{r}_{k+N} \end{bmatrix} \in \mathbb{R}^{Nm \times 1}, \tilde{\mathbf{x}} = \begin{bmatrix} \tilde{\mathbf{x}}_{k+1} \\ \tilde{\mathbf{x}}_{k+2} \\ \vdots \\ \tilde{\mathbf{x}}_{k+N} \end{bmatrix} \in \mathbb{R}^{N(n+m) \times 1}, \Delta \mathbf{u} = \begin{bmatrix} \Delta \mathbf{u}_k \\ \Delta \mathbf{u}_{k+1} \\ \vdots \\ \Delta \mathbf{u}_{k+N-1} \end{bmatrix} \in \mathbb{R}^{Nm \times 1},$$

with  $\tilde{\mathbf{x}}_k \in \mathbb{R}^{(n+m) \times 1}$  as the current state so that (4.20) may be written as

$$J = \min_{\Delta \mathbf{u}_k} \frac{1}{2} \tilde{\mathbf{x}}^\top \bar{\mathbf{V}} \tilde{\mathbf{x}} - \mathbf{r}^\top \bar{\mathbf{T}} \tilde{\mathbf{x}} + \frac{1}{2} \Delta \mathbf{u}^\top \bar{\mathbf{W}} \Delta \mathbf{u} + c_1 \quad (4.21)$$

where

$$\bar{\mathbf{V}} = \begin{bmatrix} \tilde{\mathbf{C}}^\top \mathbf{V} \tilde{\mathbf{C}} & & \dots & \mathbf{0} \\ & \tilde{\mathbf{C}}^\top \mathbf{V} \tilde{\mathbf{C}} & & \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \dots & & \tilde{\mathbf{C}}^\top \mathbf{Z} \tilde{\mathbf{C}} \end{bmatrix}, \bar{\mathbf{T}} = \begin{bmatrix} \mathbf{V} \tilde{\mathbf{C}} & \dots & \mathbf{0} \\ & \mathbf{V} \tilde{\mathbf{C}} & & \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \dots & & \mathbf{Z} \tilde{\mathbf{C}} \end{bmatrix},$$

$$\bar{\mathbf{W}} = \begin{bmatrix} \mathbf{W} & \dots & \mathbf{0} \\ & \mathbf{W} & & \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \dots & & \mathbf{W} \end{bmatrix},$$

with  $\bar{\mathbf{V}} \in \mathbb{R}^{N((n+m) \times (n+m))}$ ,  $\bar{\mathbf{T}} \in \mathbb{R}^{N(m \times (n+m))}$  and  $\bar{\mathbf{W}} \in \mathbb{R}^{N(m \times m)}$ . We want to write (4.21) in terms of the change in input  $\Delta \mathbf{u}_k$  and the present state  $\tilde{\mathbf{x}}_k$  rather than the future states. If we choose a horizon period  $N = 4$ , the corresponding states for the whole horizon period that is,  $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \tilde{\mathbf{x}}_3$  and  $\tilde{\mathbf{x}}_4$  may be computed by using (4.19) (see Appendix C.2.2) and we can compactly represent the state space system for the  $N$  future states in terms of the current state  $\tilde{\mathbf{x}}_k$  as

$$\tilde{\mathbf{x}} = \bar{\mathbf{A}} \tilde{\mathbf{x}}_k + \bar{\mathbf{B}} \Delta \mathbf{u} \quad (4.22)$$

with  $\tilde{\mathbf{x}} = [\tilde{x}_1 \ \tilde{x}_2 \ \tilde{x}_3 \ \tilde{x}_4]^\top$ ,  $\Delta \mathbf{u} = [\Delta u_0 \ \Delta u_1 \ \Delta u_2 \ \Delta u_3]^\top$ ,

$$\bar{\mathbf{A}} = \begin{bmatrix} \tilde{\mathbf{A}}_1 \\ \tilde{\mathbf{A}}_2 \tilde{\mathbf{A}}_1 \\ \tilde{\mathbf{A}}_3 \tilde{\mathbf{A}}_2 \tilde{\mathbf{A}}_1 \\ \tilde{\mathbf{A}}_4 \tilde{\mathbf{A}}_3 \tilde{\mathbf{A}}_2 \tilde{\mathbf{A}}_1 \end{bmatrix} \text{ and } \bar{\mathbf{B}} = \begin{bmatrix} \tilde{\mathbf{B}} & \dots & \dots & \mathbf{0} \\ \tilde{\mathbf{A}}_2 \tilde{\mathbf{B}} & \tilde{\mathbf{B}} & & \vdots \\ \tilde{\mathbf{A}}_3 \tilde{\mathbf{A}}_2 \tilde{\mathbf{B}} & \tilde{\mathbf{A}}_3 \tilde{\mathbf{B}} & \tilde{\mathbf{B}} & \\ \tilde{\mathbf{A}}_4 \tilde{\mathbf{A}}_3 \tilde{\mathbf{A}}_2 \tilde{\mathbf{B}} & \tilde{\mathbf{A}}_4 \tilde{\mathbf{A}}_3 \tilde{\mathbf{B}} & \tilde{\mathbf{A}}_4 \tilde{\mathbf{B}} & \tilde{\mathbf{B}} \end{bmatrix},$$

where  $\bar{\mathbf{A}} \in \mathbb{R}^{N(n+m) \times (n+m)}$  and  $\bar{\mathbf{B}} \in \mathbb{R}^{N((n+m) \times m)}$ . These matrices are constructed at each time step based on the tricopter's  $\mathbf{A}(\dot{\phi}, \dot{\theta})$  and  $\mathbf{B}$  matrices where the  $\mathbf{A}(\dot{\phi}, \dot{\theta})$  is discretized at each time step since some of its parameters change with time. It is worth noting that the reference and states start from time period  $k + 1$  and end at  $k + N$  while the change in input starts at the current time period  $k$  and ends at  $k + N - 1$ . This is because the input in the current time step affects the states and output in the next time step. Finally, if we substitute (4.22) in (4.21) and simplify the expression ignoring constant terms (see Appendix C.2.3), we obtain the cost function as,

$$J = \min_{\Delta \mathbf{u}_k} \frac{1}{2} \Delta \mathbf{u}^\top \left[ \bar{\mathbf{B}}^\top \bar{\mathbf{V}} \bar{\mathbf{B}} + \bar{\mathbf{W}} \right] \Delta \mathbf{u} + \begin{bmatrix} \tilde{\mathbf{x}}_k^\top & \mathbf{r}^\top \end{bmatrix} \begin{bmatrix} \bar{\mathbf{A}}^\top \bar{\mathbf{V}} \bar{\mathbf{B}} \\ -\bar{\mathbf{T}} \bar{\mathbf{B}} \end{bmatrix} \Delta \mathbf{u} + c_1 + c_2 \quad (4.23)$$

which is of the form

$$\frac{1}{2} \Delta \mathbf{u}^\top \mathbf{H} \Delta \mathbf{u} + \mathbf{F}^\top \Delta \mathbf{u}$$

which can be solved in Matlab using the `quadprog` function.

## 4.6 Matlab simulation results for attitude control

This section shows the simulation results with proper illustrations, and gives a comparative study of the proposed control schemes (QFB and MPC).

### 4.6.1 Simulation results using QFB technique

The quaterion feedback control scheme in Figure 4.6 was simulated in Matlab/Simulink for 10 seconds using the identified parameters and measured arm length  $l_0 = 0.33\text{m}$ . The controller gains were computed as  $\mathbf{P}_q = 4.5\mathbf{I}_3$  and  $\mathbf{D}_w = 1.95\mathbf{I}_3$  as per the requirements. Initial conditions of  $\boldsymbol{\omega}(0)^b = [0 \ 0 \ 0]^\top$  and  $\mathbf{q}(0) = [0.8 \ 0.1701 \ -0.1671 \ 0.1077]^\top$  were set for the angular velocity and attitude respectively and the results show that the controller effectively stabilizes the attitude after approximately 4s.

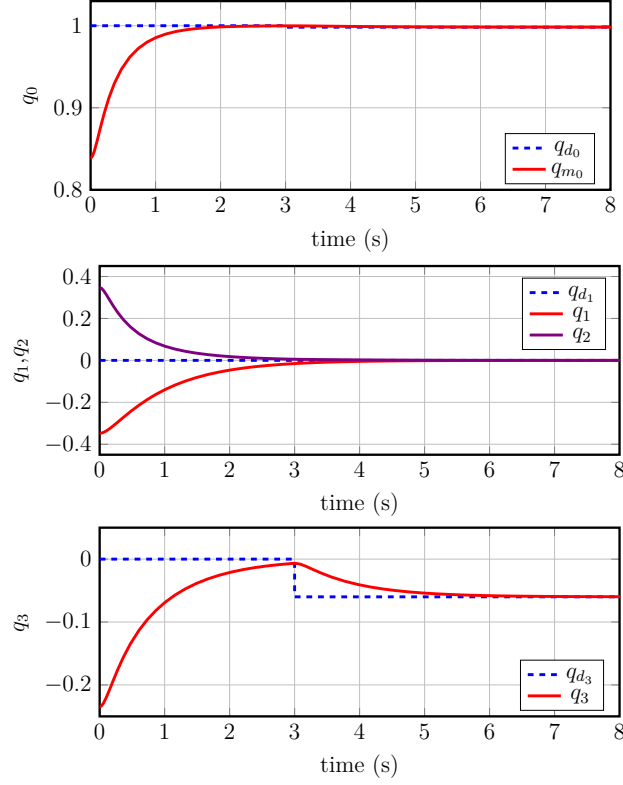


Figure 4.8: Attitude response to demanded heading of  $-0.12$  rad ( $-7$  deg.) after 3s,  $([\phi = 0, \theta = 0, \psi = -0.12]^\top \Rightarrow [q_0 = 1, q_1 = 0, q_2 = 0, q_3 = -0.059]^\top)$ .

It can be observed from Figure 4.8 that  $q_0$  which is the scalar part of the quaternion is approximately unity,  $q_1$  and  $q_2$  which are related to the roll and pitch respectively are regulated to 0, while  $q_3$  which is related to the heading tracks the value of  $-0.06$  rad (corresponding to  $\psi = -0.12$  rad) which is demanded after 3 seconds. We recover the equivalent Euler angles from the quaternions using (2.42) and as shown in Figure 4.9, the yaw angle tracks the demanded value while the roll and pitch are regulated to zero as time increases. Also, both the error and angular velocity tend to zero as time increases. That is,  $\mathbf{q}_e \rightarrow 0$  and  $\boldsymbol{\omega}^b \rightarrow 0$  as seen from Figure 4.10. The control torques are small with the control input for the demanded heading as  $\tau_\psi = -0.3 \text{ kg.m}^2/\text{s}^2$  from Figure 4.11 showing the practicability of such controller. It can also be seen that the servo angles  $\alpha_i \approx 0.05 \text{ rad}$  and the motor speeds  $\omega_i \approx 660 \text{ rad/s}$ , for  $i \in \{1, 2, 3\}$ . Equal servo angles and rotor speeds are expected since  $f_1 \cos(\alpha_1) \approx f_2 \cos(\alpha_2) \approx f_3 \cos(\alpha_3) \approx \frac{m_t g}{3}$  via symmetry in Figure 4.1. It should also be noted that all three rotors tilt to provide lateral torque to stabilize the heading rather than only the tail rotor (as in the classic case) as evident from Figure 4.11. Consequently, the servo angles are non-zero at steady hover.

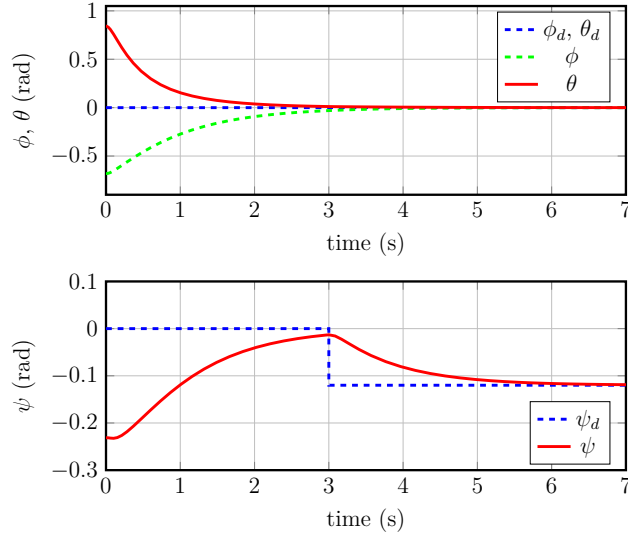


Figure 4.9: Attitude response in Euler angles  $\boldsymbol{\eta}$ ,  $([\phi = 0, \theta = 0, \psi = -0.12]^\top \Rightarrow [q_0 = 1, q_1 = 0, q_2 = 0, q_3 = -0.059]^\top)$ .

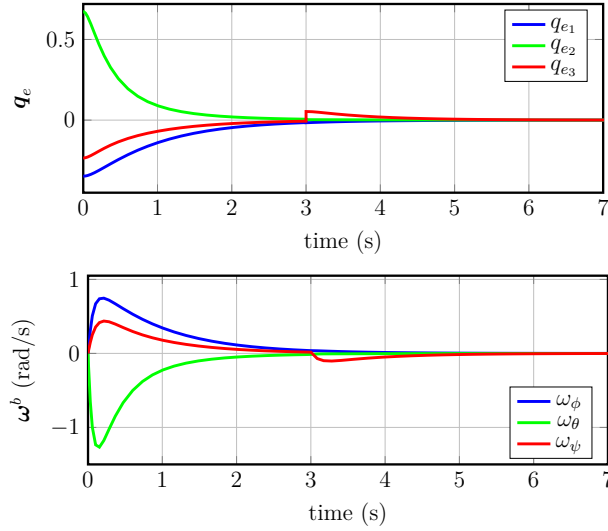


Figure 4.10: Quaternion error  $\mathbf{q}_e$  and Angular velocity  $\boldsymbol{\omega}^b$

#### 4.6.2 Comparative simulation study between QFB and MPC

The MPC scheme in Figure 4.7 was then simulated in Matlab using the tricopter's identified parameters with the `quadprog` function used to optimize the cost (4.23), and the results were compared with the results from the QFB. This comparison was done in order to analyse the performance of the linear MPC scheme which uses the quasi-LPV model, with respect to that of the QFB which is a nonlinear scheme. The controller weights were chosen as  $\mathbf{V} = \mathbf{W} = \text{diag}\{5, 5, 5\}$  and  $\mathbf{Z} = \text{diag}\{6, 6, 6\}$ . The horizon period has been noted as  $N = 4$  meaning besides the current state, 4 states in the future would be predicted by the controller at each time step.

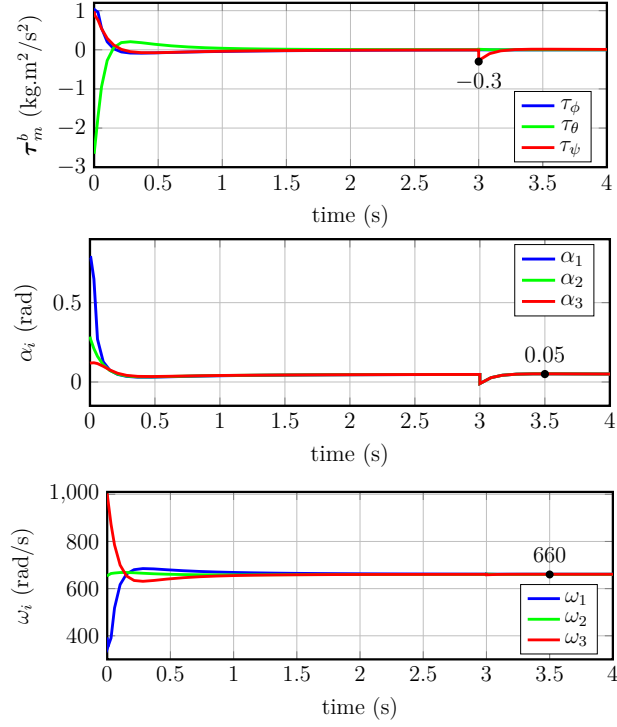


Figure 4.11: Control torques and actuator outputs,  $\tau_m^b$ ,  $\alpha_i$  and  $\omega_i$  stand for torques, servo angles and motor speeds respectively.

It can be observed from Figure 4.12, that the MPC performs better than the QFB control method driving the angles to the desired values after 2 seconds, faster than the QFB. The QFB however provides a smoother signal than the MPC at earlier time steps but it should be noted that the choppy MPC signals occur in very small magnitude in the order of approximately 0.05 rad and hence does not negatively affect the overall control or tricopter system. Furthermore, as MPC is a discrete control method, we would expect such a response [93]. Comparing the control actions between the two control techniques shows that the MPC scheme provides cheaper control compared to the QFB as seen from Figure 4.13. The peak control torques for the demanded yaw reference are -0.55 and -0.3 kg.m<sup>2</sup>/s<sup>2</sup> using the QFB, -0.2 and -0.04 kg.m<sup>2</sup>/s<sup>2</sup> when using the MPC. The same situation can be observed for the control torques for the roll and pitch channels whose angles are regulated to 0 rad respectively. This is because unlike the QFB, the MPC controller minimizes the cost function to obtain the minimal inputs required for each channel, while at the same time minimizing the state variables. It is interesting to note that although a linear MPC is used as compared with the nonlinear quaternion feedback scheme, the MPC still performs better since it makes use of a LPV model of the system which takes into account the change in the model

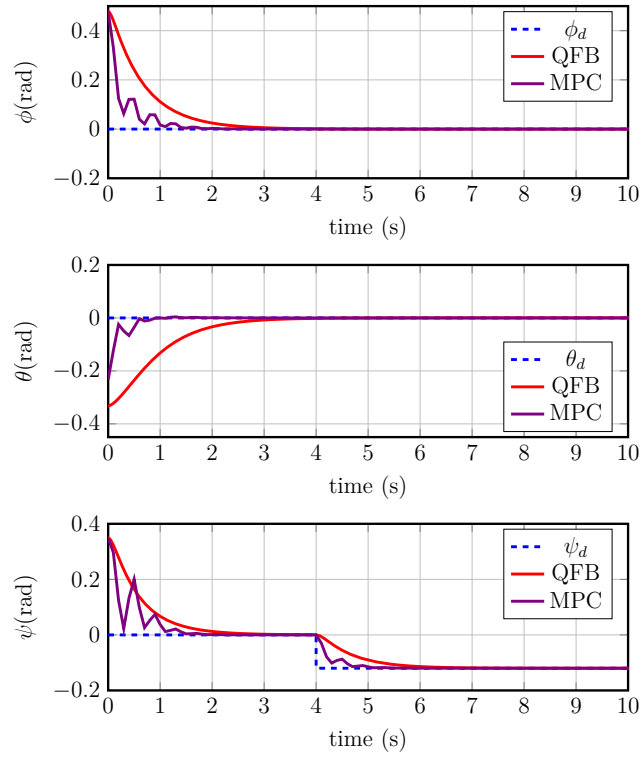


Figure 4.12: Attitude response to demanded heading of -0.14 rad with QFB and MPC

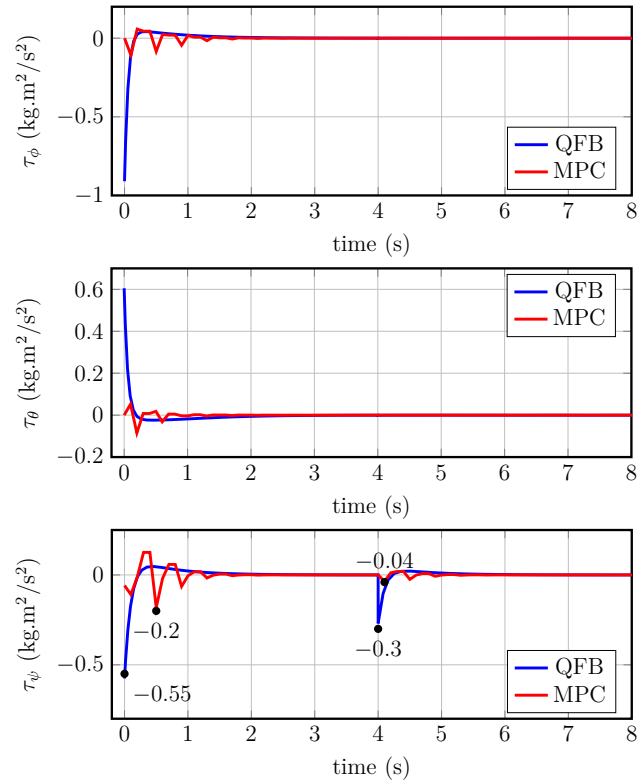


Figure 4.13: Comparing control torques between QFB and MPC schemes

parameters at each time step as seen in the varying  $\mathbf{A}$  system matrix. Like the control inputs, the generated motor speeds for the MPC are smaller compared to those of the

QFB as observed from Figure 4.14.

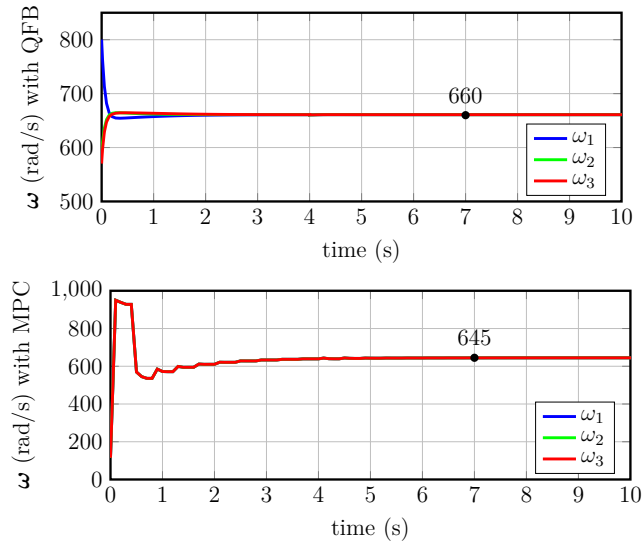


Figure 4.14: Comparing motor speeds between QFB and MPC schemes

The steady state speed from each motor can be observed to be 660 rad/s when using the QFB controller and 645 rad/s when using the MPC indicating a drop in demanded motor speed of about 15 rad/s which highlights the benefit of the MPC over the QFB scheme. One physical advantage of this is that lower motor speeds mean less power usage which in turn leads to longer flight times. Also, Figure 4.15 shows that there is no significant difference in the servo angles at steady state when using either the QFB or the MPC scheme but it can be observed that the peak of the changes in servo angles is higher when using the QFB than when using the MPC scheme. Although an in-depth robustness analysis from a theoretical standpoint has not been carried out, it is necessary to mention that this MPC scheme does not have any guaranteed stability margins. The stability and robustness depends mainly on the prediction horizon and the specific properties of the system under consideration.



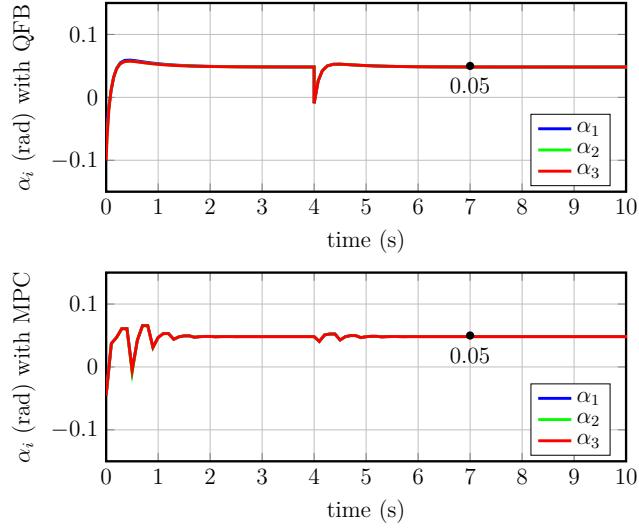


Figure 4.15: Comparing servo angles between QFB and MPC schemes

## 4.7 6-DOF control of tricopter

In addition to attitude control, this section describes the design of position controllers to perform lateral and longitudinal motions for the proposed tricopter, as a way of showing that the multi-tilt tricopter can achieve full 6-DOF control.

As noted in Remark 4.1, one important benefit of the proposed tricopter airframe compared to the classic one [75] or other common multicopters like the quadcopter [94], is that it provides the opportunity to directly and independently obtain the forces  $F_x$  and  $F_y$ , and hence control the longitudinal and lateral positions respectively of the tricopter, without changing the attitude. Consequently, full 6-DOF control is possible. In the practical sense, this position control is achieved by changing the servo angles  $\alpha_i$ ,  $i \in \{1, 2, 3\}$  thereby directing the thrust from the rotors towards the desired direction of motion. This capability is demonstrated in simulation by designing position controllers to compute  $F_x$  and  $F_y$  based on linearising the tricopter's translational equations (3.9) by feedback. It is assumed that for motion in the  $z$  direction (altitude),  $\Sigma F_x = 0$ ,  $\Sigma F_y = 0$ , and  $\Sigma F_z \neq 0$ . Similarly, for longitudinal motion,  $\Sigma F_x \neq 0$ ,  $\Sigma F_y = 0$ ,  $\Sigma F_z \neq 0$  and for lateral motion,  $\Sigma F_x = 0$ ,  $\Sigma F_y \neq 0$ ,  $\Sigma F_z \neq 0$ . Note that for lateral and longitudinal motions, we need the vertical thrust to maintain the altitude of the drone hence  $F_z \neq 0$ . Applying this information in (3.9), the translation equations for the

tricopter become,

$$\ddot{x} = \frac{1}{m_t} \left[ (c_\theta c_\psi) F_x + (c_\phi s_\theta c_\psi + s_\phi s_\psi) F_z \right], \quad (4.24)$$

$$\ddot{y} = \frac{1}{m_t} \left[ (s_\phi s_\theta s_\psi + c_\phi c_\psi) F_y + (c_\phi s_\theta s_\psi - s_\phi c_\psi) F_z \right], \quad (4.25)$$

$$\ddot{z} = \frac{1}{m_t} \left[ (c_\phi c_\theta) F_z \right] + g. \quad (4.26)$$

Let us first consider the  $z$  axis, and define the tracking errors as,  $e_z = z_d - z$ ,  $\dot{e}_z = \dot{z}_d - \dot{z}$  and  $\ddot{e}_z = \ddot{z}_d - \ddot{z}$ , then a linear control  $v_z = \ddot{e}_z$  based on pole placement is used to stabilize the error dynamics to ensure tracking and closed-loop stability hence

$$v_z = -k_{z2} \dot{e}_z - k_{z1} e_z$$

where  $k_{z1}$  and  $k_{z2}$  are chosen such that the characteristic polynomial  $s^2 + k_{z2}s + k_{z1} = 0$  is Hurwitz. Note that  $z_d$  is the desired altitude. Then from  $v_z = \ddot{z}_d - \ddot{z}$ , we get  $\ddot{z} = \ddot{z}_d - v_z$  and by substituting this in (4.26) we obtain the control input  $F_z$  as,

$$F_z = \frac{m_t(\ddot{z}_d - v_z - g)}{c_\phi c_\theta}. \quad (4.27)$$

Following a similar process, the control forces for the lateral and longitudinal motions are obtained as,

$$F_x = \frac{m_t(\ddot{x}_d - v_x) - (c_\phi s_\theta c_\psi + s_\phi s_\psi) F_z}{c_\phi c_\theta}, \quad (4.28)$$

$$F_y = \frac{m_t(\ddot{y}_d - v_y) - (c_\phi s_\theta s_\psi - s_\phi c_\psi) F_z}{s_\phi s_\theta s_\psi + c_\phi c_\theta} \quad (4.29)$$

where  $v_x = \ddot{e}_x$ ,  $v_y = \ddot{e}_y$ .

**Remark 4.4.** Note that for all simulations, the actuator outputs from the mixer are computed from (4.7) as  $\mathbf{\Omega} = \mathbf{M}^{-1}\mathbf{U}$  where  $\omega_i = \sqrt[4]{\Omega_i^2 + \Omega_{i+3}^2}$  and  $\alpha_i = \text{atan}(\frac{\Omega_i}{\Omega_{i+3}})$  for  $i \in \{1, 2, 3\}$ . In the earlier simulations (QFB and MPC) where only attitude control is considered, the control forces are arbitrarily set as  $F_x = F_y = 0$  and  $F_z = -m_t g \neq 0$  in order to simulate the hover position. In the simulation of the position controllers however,  $F_x \neq 0$  and  $F_y \neq 0$  as expected due to (4.28) and (4.29).

For the three position axes, the poles are chosen at  $s = -1, -2$  which yields the gain  $k = [2 \ 3]$ . The QFB scheme in Figure 4.6 is then simulated but with the position controllers now included to achieve 6-DOF control hence full actuation. The reference

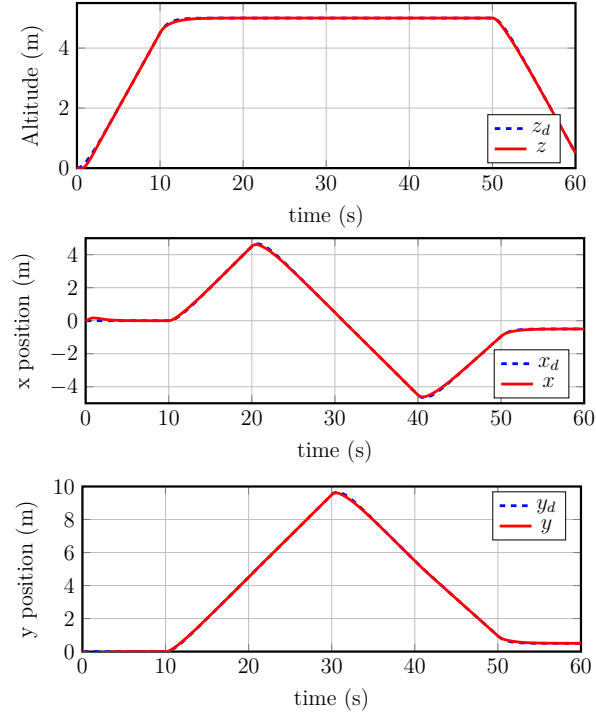


Figure 4.16: Position of tricopter using translational forces only without changing attitude

attitude is zero that is,  $\mathbf{q}_d = [1 \ 0 \ 0 \ 0]^\top$  and a 60 seconds trajectory is demanded where the drone takes off to an altitude of 5m, follows a rectangular path and then returns back to the ground level. From Figures 4.16 and 4.17, the output positions track the desired values and corresponding trajectory.

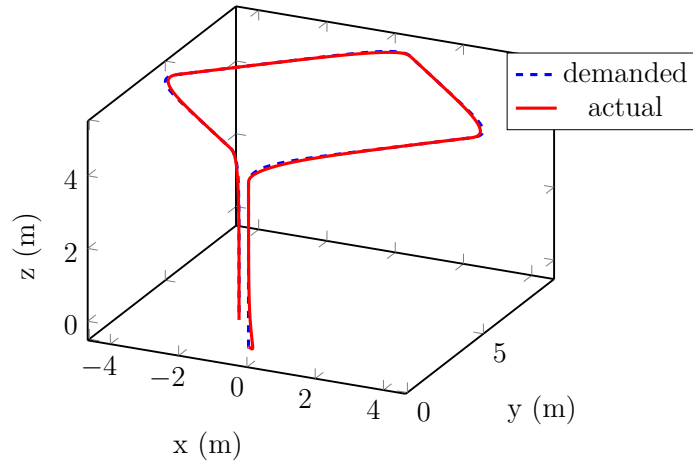


Figure 4.17: Tricopter response to rectangular trajectory with motion achieved using translational control forces only.

This motion is made possible using the forces  $F_x$ ,  $F_y$  and  $F_z$  only, unlike a typical multicopter where position control is achieved by changing the attitude based on

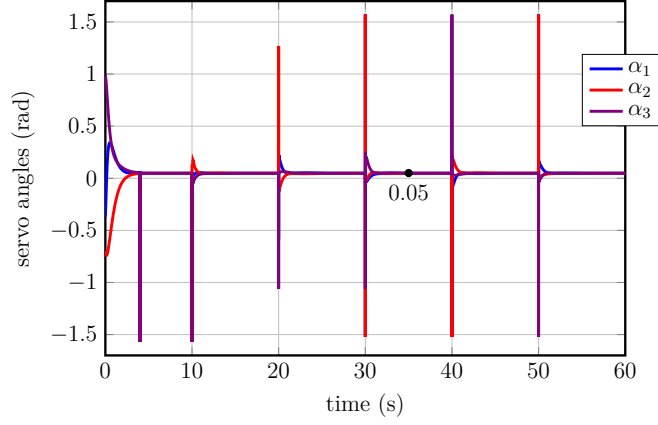


Figure 4.18: Tilt angles applied to the servo motors to enable motions in  $x$  or  $y$  directions.

generated roll, pitch and yaw reference values which are computed from the desired  $x$  and  $y$  positions as in [95–97]. The tilting angles  $\alpha_i, i \in \{1, 2, 3\}$  are given in Figure 4.18. It can be observed that these angles change as the demanded position changes, before returning to their steady state value of 0.05 rad. The change in these angles results in the respective motor being tilted clockwise or counter-clockwise thereby directing the thrust of that motor from the vertical axis towards the desired direction of motion. It can also be observed that the peak value of the tilt angle is approximately 1.57rad ( $90^\circ$ ) since the servo motor in the physical hardware is limited to the range  $[-\frac{\pi}{2}, \frac{\pi}{2}]$ . Although, this has not caused significant issues in the simulation, it is worth mentioning that actuator saturations in real systems can cause instability. We normally want to avoid tilt angle demands that reach the limits of the servo motor as this may lead to wind-up which if left unchecked, can destabilize the system.

For completeness, the attitude response is given in Figure 4.19 (in terms of euler angles) for the first 10 seconds, and it can be observed that the roll, pitch and yaw angles are regulated to 0 by the QFB controller as desired.

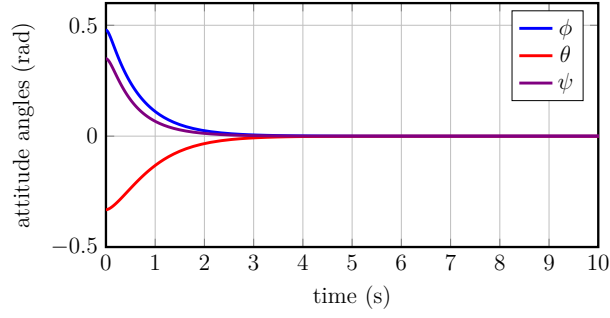


Figure 4.19: Attitude response to reference  $\mathbf{q}_d = [1 \ 0 \ 0 \ 0]^\top$  ( $\boldsymbol{\eta} = [0 \ 0 \ 0]^\top$ ) with position control for 6-DOF control.

## 4.8 Experiments

This section describes the experimental aspects of this work. Firstly, a description of the hardware components used in realising the proposed airframe is provided. The developed tilt mechanism which allows for a greater tilting range is then discussed followed by a description of the control allocator (mixer) to achieve independent position control and its software implementation. Finally, some experimental tests are done to confirm that the mixer enables the tricopter to vector its thrusts from the rotors onto the longitudinal and lateral axes as required, to achieve independent position control on these axes.



Figure 4.20: Tricopter platform on test stand

### 4.8.1 Platform Description

The proposed multi-tilt tricopter UAV platform is shown in earlier figures and in Figure 4.20 where it is affixed to a test stand. The platform consists of three identical arms, with a fixed pitch propeller driven by a Brushless DC motor mounted at the end of each arm. Each motor is mounted on a MG996R Tower-pro servo-motor which controls the tilting of the motors in the range  $\alpha_i \in (-\frac{\pi}{2}, \frac{\pi}{2})$  for  $i \in \{1, 2, 3\}$  to achieve lateral and longitudinal motion as well as heading control. It is worth mentioning that the airframe is adapted from the HobbyKing Talon tricopter airframe where the classic tail servo tilting mechanism is extended to the two front rotors so that all three rotors can tilt. One problem is that the classic Talon servo tilting mechanism was designed for tilting in the range  $\alpha \in (-\frac{\pi}{4}, \frac{\pi}{4})$  which limits the agility of the airframe. To solve this problem, a custom tilting mechanism is designed as in Figure 4.21 based on a standard hub shaft servoblock from ServoCity with 25T spline.

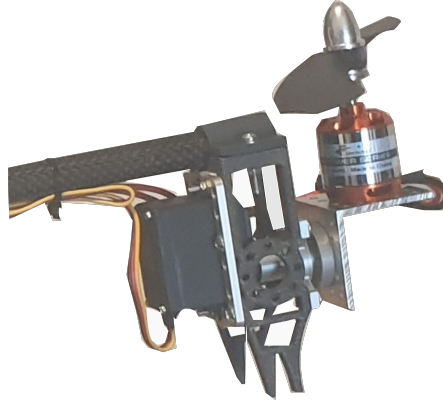


Figure 4.21: Tilting mechanism assembly.

Top and bottom mounts with landing gears are made to hold the tricopter arms in place, an L-shaped bracket on which the rotor sits is then attached to the hub shaft while the MG996R servo is attached on the other end of the hub shaft. As the servo rotates, the hub shaft rotates the assembly thereby rotating the rotor through the desired angle. This mechanism allows for tilting in the range  $[-\pi/2, \pi/2]$  (which allows for more agility in the airframe). The *Pixhawk* autopilot [98] which runs a 32bit processor with 256KB RAM, is used as the flight controller. The PX4 firmware which is compatible with Pixhawk is used to run the guidance and control algorithms and QGroundControl [99] is used as the ground station software for setup and calibration. See Appendix B.2 for full specifications. This platform was developed in the Control,

Dynamics and Robotics (CDR) lab at the University of Manchester. The platform was fixed to a safe test stand for indoor tests.

### 4.8.2 Mixer for position control

The control allocation (or Mixer) computes the actuator inputs based on the geometry of the UAV to be controlled. The tricopter under investigation has 6 control inputs  $\mathbf{U}$  and 6 actuators (3 servos and 3 motors) from (4.7). The problem however is that only four  $[F_z, \tau_\phi, \tau_\theta, \tau_\psi]^\top$  (the altitude, roll, pitch and yaw) of the six degrees-of-freedom can be controlled by any typical radio transmitter. So in line with section 4.7, a direct actuation method using the mixer is proposed, where the servos are used to direct the thrust of the vehicle in the  $x$  or  $y$  direction to create longitudinal or lateral motion as depicted in Figure 4.22. That is, forces for  $F_x$  and  $F_y$  are manually achieved using

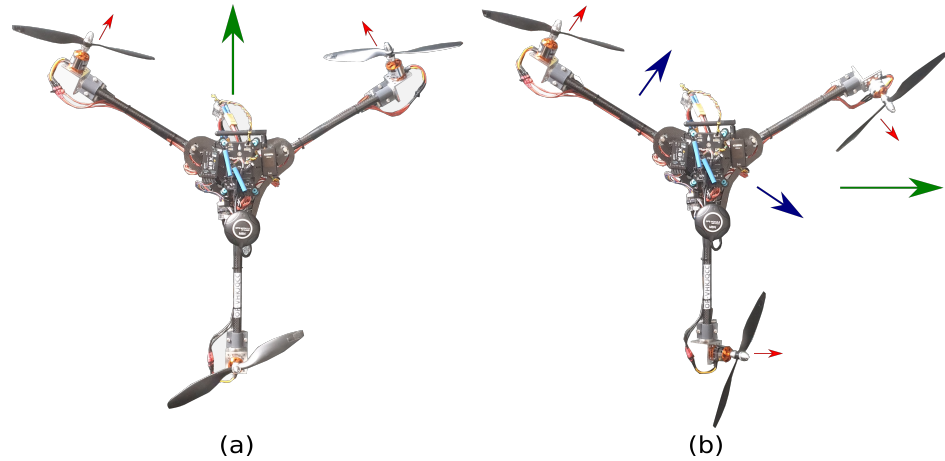


Figure 4.22: Thrust vectoring via direct actuation

inputs from specific knobs on the transmitter. We can explain this concept from Figure 4.22. If we view each rotor from the outer edge of the respective arm, then for  $+x$  motion, servo 1 rotates clockwise and servo 2 rotates counter-clockwise so that motors 1 and 2 are tilted forwards thereby directing the thrust from the motors towards the  $+x$  axis. Note that in Figure 4.22(a), the shorter arrows represent the direction of rotation of individual servos while the longer arrow represents the resultant direction of motion of the tricopter. Then for  $+y$  motion as depicted in Figure 4.22(b), both servos 1 and 2 rotate counter-clockwise when each of them are viewed from the outer edge of their respective arms, while servo 3 rotates clockwise. Here the rotation of servos 1 and 3 together tilts motors 1 and 3 such that the thrust from these motors is directed

diagonally and opposite to the arm (axis) of rotor 2. At this state, the tricopter will move diagonally between the  $-x$  and  $+y$  axes at an angle of approximately  $60^\circ$  from the  $-x$  axis. But we desire a straight-line motion in  $+y$  direction and the correction needed to achieve this is delivered by the thrust created due to the tilting of motor 2 in the counter-clockwise direction so that the resultant motion of the tricopter is in the  $+y$  direction. Note here that the intermediate thrusts are indicated by the medium sized arrows in Figure 4.22(b) while the resultant motion of the tricopter is indicated with the longer arrow. The attitude in the mixer is specified similar to a standard multicopter, where roll and pitch motions are controlled via differential speeds of the rotors as defined by (4.6). The difference is that in the proposed airframe, there are three servos which all contribute to yaw control rather than only the tail servo as with the classic tricopter [75].

### 4.8.3 Mixer Implementation

From the controller diagrams for multicopters in PX4 documentation [19], we see that in the inner loop, the attitude controller feeds setpoints to the angular rate controller which then outputs normalized torques to the mixer. Also, the outer loop position controller outputs a normalized thrust to the mixer. No part of the control scheme considers the precise geometry of the UAV platform to be controlled. The platform's characteristics are present only in the Mixer block. Hence, in the initial case, the existing PX4 control algorithms are used and only the mixing algorithm is modified to reflect the geometry in Figure 4.1 that is, the relationship described in (4.7). Consequently, we only need to tune the existing controllers to operate the airframe. Using the identified parameters, a multicopter-type PX4 mixer for the geometry in Figure 4.1 is developed, to generate PWMs to the motors for attitude mixing. For direct actuation in  $x$  and  $y$ , additional mixers are written for the servos. As an example, a mixer block is given in Figure 4.23 for servo 1 (see Appendix E for complete mixers) where in line 31, `M:` defines the number of inputs to this actuator (3 inputs/scalers) since servo 1 is requested for yaw, longitudinal and lateral motions. The next three lines define the output scalers.

So considering line 32 from left to right, `S:` takes an input from control group `#0` (which is flight control in PX4) and the yaw input given by the index `#2` (`#0`=roll,



```

30  # Servo 1
31  M: 3
32  S: 0 2 -10000 -10000 0 -10000 10000
33  S: 3 5 10000 10000 0 -10000 10000
34  S: 3 6 -10000 -10000 0 -10000 10000

```

Figure 4.23: Pseudo-code for AUX mixer implementation

#1=pitch, #2=yaw in PX4 flight control options). It scales the yaw input by 1 and reverts the sign #-10000 -10000, it applies no offset #0 and outputs the full range #-10000 10000 (that is,  $[-1, +1]$ ). Note that the values stored in the PX4 mixer definition are scaled by a factor of 10000 that is,  $[-1, +1]$  is encoded as  $[-10000, 10000]$ .

Similarly in line 34, S: takes an input from control group #3 (manual passthrough) and from a specific knob mapped via AUX2 RC channel #6 (input is direct from RC transmitter rather than flight controller, #5=RC\_MAP\_AUX1 and #6=RC\_MAP\_AUX2 in PX4 manual passthrough options). It scales the input by 1 and reverts the sign, applies no offset and outputs the full range  $[-1, +1]$ . The signs of the scalers determine whether the servo rotates clockwise or counter-clockwise when it receives an input from either the flight controller or transmitter. It should be noted that RC\_MAP\_AUX1 and RC\_MAP\_AUX2 are parameters in the PX4 firmware and they were configured to specific knobs (channels) on a TaranisX9D transmitter. To test the mixer, the PX4 firmware is rebuilt with the modified airframe setup and flashed on the Pixhawk.

#### 4.8.4 Experimental Results

The tricopter platform affixed to the test stand was armed (powered on) and allowed for about a minute before the tests were started. Note that the tests were done in closed-loop. As described in subsection 4.8.2, the motion in Figure 4.22(a)(positive  $x$  direction) is achieved by tilting servos 1 and 2 clockwise and counter-clockwise respectively. And this is implemented using a knob on the radio transmitter mapped to RC\_MAP\_AUX1 as noted in subsection 4.8.3, while the angle of servo 3 stays close to the servo center. This servo tilting (and resultant rotor tilting) was achieved by turning the knob (dial) on the radio transmitter (which was mapped to the  $x$  axis) in the direction of positive PWM resulting in the two front rotors tilting forwards as desired. The data was logged to an SD-card on the PX4 flight controller and the

motion of the servos is depicted in the top plot of Figure 4.24 between  $t = 68$  and  $t = 70$  seconds.

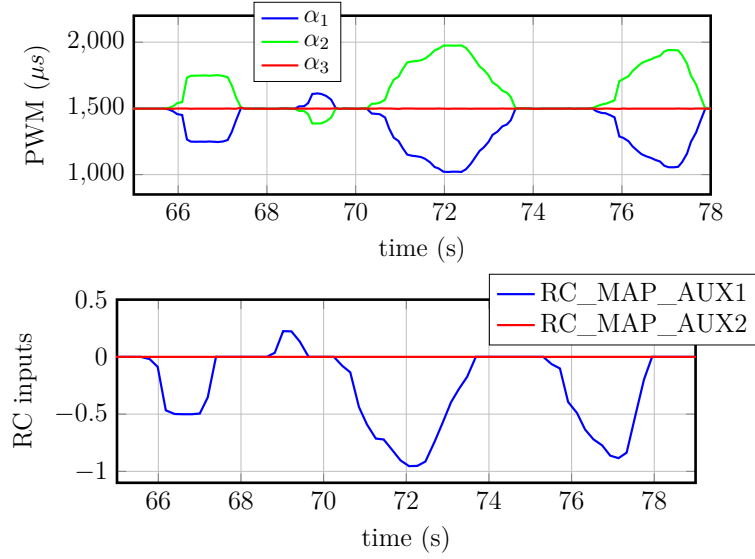
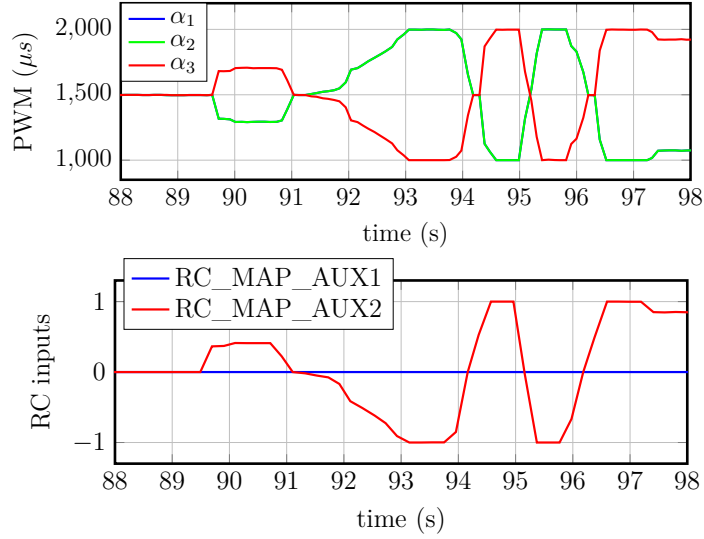


Figure 4.24: Servo outputs when using direct actuation for  $x$  motion

Between  $t = 66$  and  $t = 68$  seconds or  $t = 70$  and  $t = 74$  seconds, we have the reverse case, where rotor 1 tilts counter-clockwise and rotor 2 tilts clockwise with rotor 3 staying close to the center, and this corresponds to motion in the negative  $x$  direction. The bottom plot shows the corresponding radio transmitter (RC) signals. It can also be observed from the top plot of Figure 4.24 that servo 3 ( $\alpha_3$ ) has output approximately  $1500\mu s$  which is close to the standard servo center PWM value of  $1520\mu s$ . Similarly, to achieve the motion in Figure 4.22(b) (positive  $y$  direction), servos 1 and 2 are tilted counter-clockwise while servo 3 tilts clockwise via a knob mapped to RC\_MAP\_AUX2. This servo motion was achieved by turning the knob (dial) on the radio transmitter (which was mapped to the lateral axis) in the direction of positive PWM yielding the desired tilting of the servos (and consequently the rotors), and this is observed in the top plot of Figure 4.25, between  $t = 89$  and  $t = 91$  seconds, for example.

Figure 4.25: Servo outputs when using direct actuation for  $y$  motion

## 4.9 Summary

In this chapter, the hardware for a tricopter where all rotors can independently tilt is developed together with its mathematical model. The parameters of the model were identified via experimental methods. A quaternion-based feedback controller was then proposed to stabilize the attitude of the tricopter and this was done with the attitude represented as unit quaternions to avoid gimbal lock. A linear MPC scheme based on a linear-parameter-varying model of the tricopter was also proposed to stabilize the attitude. These control schemes have been implemented in simulation using the identified parameters and a comparative study was carried out to reflect on the performance of the control schemes. The simulation results reported show the effectiveness of the proposed QFB and MPC control schemes, their similarities and differences. The LPV-MPC showed superior performance to the quaternion feedback scheme and this is expected since MPC is based on optimizing a cost function. The capability of the proposed tricopter to directly control the motions in  $x$  and  $y$  directions is demonstrated via simulations, and then a control allocator/mixer which is implemented using the PX4 framework is proposed and this is validated in indoor experiments. Also, the tilting limit of the airframe tilt mechanism is relaxed by developing a custom tilt mechanism that allows for tilting in a wider range of  $[-\pi/2, \pi/2]$  which increases the capabilities of the airframe. The experimental results have highlighted the benefits of the proposed control allocation (and airframe) and provided an important base for

further research into this tricopter problem.

## Chapter 5

# A negative imaginary robust formation control scheme for a group of networked tricopters over inner-loop sliding-mode control

All the material in this chapter was submitted for publication in [100].

### 5.1 Introduction

Cooperative control of multicopter Unmanned Aerial Vehicles (UAVs) has gained significant attention both from academia and industry due to the improved reliability and efficiency it offers, leading to a vast number of applications [101] including search and rescue [102], pipeline inspection and surveillance [103] among others. This attention has led to more efficient configurations and innovations in terms of size, flight range, airframe configuration and other factors (see [104] for a detailed review of multicopters). An example of an innovative platform with significant properties is the multi-tilt tricopter [20] with three rotors which can independently tilt, which offers greater agility and flexibility (same tricopter considered in chapter 4). In contrast to the more common quadcopter [94] where trajectory tracking is achieved by manipulating the attitude, the multi-tilt tricopter which is considered in this work is able to achieve simultaneous independent attitude and trajectory tracking (full 6 degrees

of freedom control), due to its airframe configuration. In other words, it is possible to translate in the lateral and longitudinal directions without changing the attitude, thereby overcoming the limitation of quadcopters. This serves as the justification and motivation for investigating the cooperative control of the multi-tilt tricopter rather than the quadcopter, thus creating more possibilities when performing specialized and more complex tasks. One of the most investigated coordination problems of multiagent systems (MAS) is the formation control problem with the key concern being the development of control strategies to achieve group formation tracking. Consequently, several approaches [30, 36, 37] have been proposed to solve the formation control problem including the more recent Negative Imaginary (NI) approach [41, 42]. NI systems theory was originally inspired by the ‘positive position feedback control’ of highly resonant mechanical systems [47]. It initially drew attention as a control technique for vibration control of flexible structures [49], large space structures and robotic manipulators [48], and later saw applications in nano-positioning [105], vehicle platooning [51] and control of networked MAS [31, 58]. The key feature driving this growth is its simple internal stability condition that says a necessary and sufficient condition for the internal stability of a positive feedback interconnection of NI and SNI systems, say  $M(s)$  and  $N(s)$ , is  $\lambda_{\max}[N(0)M(0)] < 1$  [47, 54], which is a condition that depends on the loop gain at zero frequency only. In addition, the NI theory offers a stand-alone robust control analysis and synthesis framework [106, 107]. Ref. [41] presents a framework for swarm multiagent systems based on the relative-position output feedback consensus combined with adaptive strictly negative imaginary controllers, leveraging the learning capability of artificial neural networks. A case study of two quadcopters moving together while carrying a dynamic load is used to validate the proposed method in experiments. NI theory is utilized in [55] to design a consensus-based formation control framework for a multi-vehicle system together with an obstacle detection and avoidance algorithm. The performance of the obstacle avoidance technique proposed is compared to that of [56] which is based on artificial potential field function control technique. In [56] and [57], authors have utilised the results of [31] and [32] to develop a particular consensus and formation control framework for a group of heterogeneous autonomous vehicles facilitating time-invariant switching. Also, a rendezvous control scheme for a group of two-wheeled mobile robots connected via a directed and balanced graph has been

designed in [58] using NI theory. The concept of utilising the NI-SNI closed-loop stability result to develop a cooperative control scheme stems from the fact that a certain class of systems can be modelled as a group of networked single/double integrator agents typically by feedback linearization, which inherently satisfies the NI property with poles at the origin.

Consequently, a cooperative control scheme can be deployed as the outer-loop controller. The nonlinear model of the multi-tilt tricopter system using force and torque kinematic analysis and subsequent dynamic modelling is first presented. Different from [20] where feedback linearization has been employed, sliding mode technique is used for designing control laws to linearize the inner-loop of the nonlinear multi-tilt tricopter UAV. The SMC technique was used because it is a nonlinear control method and therefore could be applied directly to the tricopter which is a highly nonlinear system. Secondly, it is straight-forward to implement and is known to have inherent robustness properties which is desirable as we want to avoid any issues in the inner-loop since the outer-loop cooperative controller depends on it. As a result of using the SMC technique, an SMC-linearized closed-loop system with six inputs and outputs is obtained, corresponding to the tricopter's cartesian position and attitude. To characterize the SMC-linearized inner-loop as an NI system, a closed-loop system identification algorithm that guarantees that the resultant model is NI, is proposed. The algorithm presented exploits the classic subspace method but in continuous time by use of the Laguerre filter [59] in the identification process, and its advantages and simplicity are highlighted. This algorithm is then applied to the identification of NI models for all channels of the SMC-linearized multi-tilt tricopter and frequency responses of the identified models are used to verify that the models exhibit the NI property. An output feedback distributed SNI+VSP control law is then designed for achieving robust cooperative control combining both leader-following consensus and formation control. Rather than the Lyapunov-based approach, a methodology proposed in [42, 70] is utilized in proving the convergence of the control problem which encompasses formation control and cooperative tracking, exploiting the characteristics of the eigenvalue loci of networked NI and SNI systems. Finally, a simulation case study involving a group of six SMC-linearized multi-tilt tricopter agents is provided to demonstrate the usefulness and effectiveness of the proposed scheme.

The key contributions of this chapter to the state-of-the-art are as follows:

- Development of a closed-loop system identification algorithm that guarantees that the resultant model is NI. The algorithm presented exploits the classic subspace method but in continuous time by use of the Laguerre filter [59] in the identification process, and its advantages and simplicity are highlighted.
- Development of closed-loop stability result for a network of NI/SNI systems which are controlled by ‘mixed’ Strictly Negative Imaginary (SNI) plus Very Strictly Passive (VSP) controllers in a negative feedback interconnection.
- Design of an output feedback distributed SNI+VSP control law (making use of the closed-loop stability result for networked NI/SNI systems) for achieving robust cooperative control combining both leader-following consensus and formation control, utilizing the characteristics of the eigenvalue loci of networked NI and SNI systems for proving the convergence of the control problem, rather than the Lyapunov-based approach.

The other technical contributions of this chapter are summarized as follows:

- Design of Sliding Mode Controller (SMC) to linearize the inner-loop of the nonlinear multi-tilt tricopter UAV, different from feedback linearization which is commonly used in such situations. This yields an SMC-linearized closed-loop system with six inputs and outputs corresponding to the multi-tilt tricopter’s cartesian positions and attitude.
- Identification of NI models for all channels of the SMC-linearized multi-tilt tricopter using the developed closed-loop identification algorithm, and frequency responses of the identified models are used to verify that the models exhibit the NI property.
- Provision of simulation case study involving a group of six SMC-linearized multi-tilt tricopter agents, achieving group formation tracking and inherent consensus tracking, demonstrating the usefulness and effectiveness of the proposed scheme.

The rest of this chapter is organized as follows: A problem formulation is provided next, then the derivation and SMC linearization of the nonlinear dynamic model



of the tricopter UAV is described in Section 5.2. The proposed closed-loop system identification algorithm is presented in Section 5.3 together with the identification of NI models for the inner-loop of the linearized tricopter UAV. Section 5.4 develops the key stability result for the SNI+VSP controller and applies it to the formation control of a network of multi-tilt tricopter systems. Section 5.5 is devoted to the presentation of the case study and simulation results, and a summary is provided in Section 5.6.

### 5.1.1 Problem Formulation

Given a multi-agent system with  $N$  nodes, distributed on a communication graph  $\mathcal{G}$  and modelled as a group of networked agents with identical dynamics, the control problem is to design distributed output feedback SNI+VSP controllers  $u_i \forall i \in \{1, 2, \dots, N\}$  in  $\mathcal{G}$  such that all nodes converge to the state trajectory of the leader node and the agents asymptotically reach the desired static formation. That is,  $\lim_{t \rightarrow \infty} (x_i(t) - x_0(t)) = 0$ ,  $\forall i \in \{1, 2, \dots, N\}$ , where  $x_0$  is the state of the leader node and  $x_i$  is the state of each individual agent.

## 5.2 Multi-tilt tricopter modelling

This section presents the model of the tricopter UAV under consideration. Note that the same model has been developed in Section 4.2 of Chapter 4. Hence, the nonlinear model of the multi-tilt tricopter from Section 4.2 is still applicable. The linearization of the model using sliding mode control technique in closed-loop is discussed next. This SMC-linearized closed-loop system will be used in later sections of this chapter.

### 5.2.1 Linearisation using Sliding Mode Control

Considering the attitude dynamics (3.10), an assumption is made that the angles  $\phi$  and  $\theta$  are small and hence  $\mathbf{\Gamma}^{-1} \approx \mathbf{I}_3$  from (2.41), such that from (3.10)  $\ddot{\boldsymbol{\eta}} = \dot{\boldsymbol{\omega}}^b$  and expanding yields

$$\ddot{\phi} = [(J_y - J_z)\dot{\theta}\dot{\psi} + \tau_\phi]/J_x, \quad (5.1)$$

$$\ddot{\theta} = [(J_z - J_x)\dot{\phi}\dot{\psi} + \tau_\theta]/J_y, \quad (5.2)$$

$$\ddot{\psi} = [(I_x - I_y)\dot{\phi}\dot{\theta} + \tau_\psi]/J_z. \quad (5.3)$$

Let us first consider the roll dynamics (5.1) and using classical methods [45, 108], define a sliding surface  $\tilde{s}_\phi = \dot{e}_\phi + \lambda_\phi e_\phi$  where  $\lambda_\phi > 0$  and  $e_\phi = \phi - \phi_d$  is the error between the actual and desired roll angles. A control law  $\dot{\tilde{s}} = -k_\phi \text{sgn}(\tilde{s}_\phi)$ ,  $k_\phi > 0$  is chosen for the attractive surface and to analyse stability, a Lyapunov candidate function  $V_\phi = \frac{1}{2}\tilde{s}_\phi^2 > 0$  is chosen so that

$$\begin{aligned}\dot{V}_\phi &= \tilde{s}_\phi \dot{\tilde{s}}_\phi = \tilde{s}_\phi (-k_\phi \text{sgn}(\tilde{s}_\phi)) = -k_\phi |\tilde{s}_\phi| \leq 0, \\ \Rightarrow \tilde{s}_\phi \left( \frac{1}{J_x} [(J_y - J_z) \dot{\theta} \dot{\psi} + \tau_\phi] - \ddot{\phi}_d + \lambda_\phi \dot{e}_\phi \right) &= -k_\phi |\tilde{s}_\phi|, \\ \Rightarrow \tau_\phi &= J_x (-k_\phi \text{sgn}(\tilde{s}_\phi) + \ddot{\phi}_d - \lambda_\phi \dot{e}_\phi) - (J_y - J_z) \dot{\phi} \dot{\psi}\end{aligned}$$

where

$$\text{sgn}(\tilde{s}) = \frac{|\tilde{s}|}{\tilde{s}} = \begin{cases} -1, \tilde{s} < 0 \\ 0, \tilde{s} = 0 \\ 1, \tilde{s} > 0 \end{cases}. \quad (5.4)$$

Since  $V_\phi > 0$  and the control law  $\dot{\tilde{s}}_\phi$  satisfies  $\dot{V}_\phi \leq 0$ , then  $\tilde{s}_\phi = 0$  is an equilibrium point and is finite-time stable [45], implying that the input  $\tau_\phi$  stabilizes the roll attitude. Similarly,

$$\begin{aligned}\tau_\theta &= J_y (-k_\theta \text{sgn}(\tilde{s}_\theta) + \ddot{\theta}_d - \lambda_\theta \dot{e}_\theta) - (J_z - J_x) \dot{\theta} \dot{\psi}, \\ \tau_\psi &= J_z (-k_\psi \text{sgn}(\tilde{s}_\psi) + \ddot{\psi}_d - \lambda_\psi \dot{e}_\psi) - (J_x - J_y) \dot{\phi} \dot{\theta}.\end{aligned}$$

For the translational dynamics (3.9), the following assumptions are made similar to those made in Section 4.7. For vertical motion,  $\Sigma F_z \neq 0$ ,  $\Sigma F_x = \Sigma F_y = 0$  since only vertical thrust is required for hovering. For longitudinal motion,  $\Sigma F_y = 0$ ,  $\Sigma F_x \neq 0$ ,  $\Sigma F_z \neq 0$  since vertical thrust is required to hold the altitude and a longitudinal force is required for motion in the  $x$  direction. And for lateral motion,  $\Sigma F_y \neq 0$ ,  $\Sigma F_x = 0$ ,  $\Sigma F_z \neq 0$  since vertical thrust is required for altitude and lateral force is required for  $y$  motion. Expanding (3.9) and applying these assumptions yields

$$\begin{aligned}\ddot{x} &= [(c_\theta c_\psi) F_x + (c_\phi s_\theta c_\psi + s_\phi s_\psi) F_z] / m, \\ \ddot{y} &= [(s_\phi s_\theta s_\psi + c_\phi c_\psi) F_y + (c_\phi s_\theta s_\psi - s_\phi c_\psi) F_z] / m, \\ \ddot{z} &= g + [(c_\phi c_\theta) F_z] / m.\end{aligned}$$

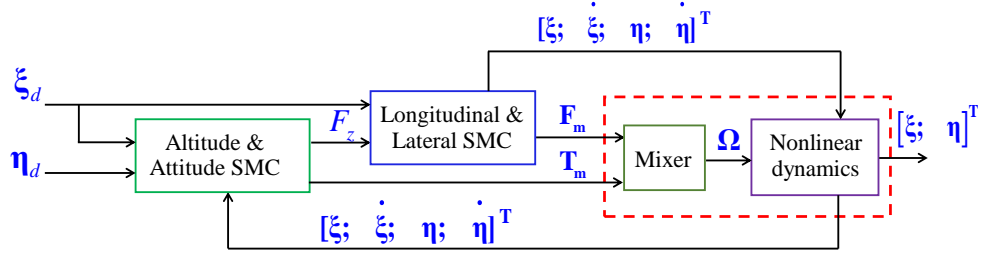


Figure 5.1: SMC control scheme for linearising tricopter dynamics

By choosing sliding surfaces  $\tilde{s}_x, \tilde{s}_y, \tilde{s}_z$  and position errors  $e_x, e_y, e_z$  similar to the attitude control case, together with matching reaching laws, the SMC linearizing control inputs for the translational dynamics are

$$F_x = \frac{m}{c_\phi c_\psi} \left( -k_x \operatorname{sgn}(\tilde{s}_x) + \ddot{x}_d - \lambda_x \dot{e}_x \right) - \frac{F_z}{c_\phi c_\psi} (c_\phi s_\theta c_\psi + s_\phi s_\psi),$$

$$F_y = \frac{m}{(s_\phi s_\theta s_\psi + c_\phi c_\psi)} \left( -k_y \operatorname{sgn}(\tilde{s}_y) + \ddot{y}_d - \lambda_y \dot{e}_y \right) - \frac{F_z}{(s_\phi s_\theta s_\psi + c_\phi c_\psi)} (c_\phi s_\theta s_\psi - s_\phi c_\psi),$$

$$F_z = \frac{m}{c_\phi c_\theta} \left( -k_z \operatorname{sgn}(\tilde{s}_z) - g + \ddot{z}_d - \lambda_z \dot{e}_z \right).$$

Using these SMC laws for both attitude and position, the scheme depicted in Figure 5.1 is then simulated using the multi-tilt tricopter's parameters given in Table 5.1 (same as those identified in Chapter 4) with  $\lambda_x = \lambda_y = \lambda_z = 1.4, \lambda_\phi = \lambda_\theta = \lambda_\psi = 1.3, k_x = k_y = 1, k_z = 1.5$  and  $k_\phi = k_\theta = k_\psi = 2.5$ , to yield a linearized closed loop with six inputs and outputs corresponding to the three positions and three attitude angles. Note that although some methods exist in the literature for choosing optimal parameters for SMC gains, the parameters used here were chosen using a heuristic approach bearing in mind that higher gains will achieve stability faster but would amplify disturbances while lower gains will reduce disturbances and sacrifice some stability. Furthermore, some approximations to discontinuities have been used in simulation to reduce the effects of chattering.

Figures 5.2 and 5.3 show the responses of the multi-tilt tricopter to a spiral trajectory with zero attitude change. It is worth recalling at this point that the tricopter under consideration can achieve independent  $x$  and  $y$  motions without changing its attitude due to its airframe configuration.

It is also evident that the 6-DOF controller from section 4.7 with a settling time of approximately 3 seconds performs better than the SMC with a settling time of approximately 5 seconds as observed from Figure 5.3. This is not a significant difference

Table 5.1: Summary of tricopter parameters

Parameter	Value
$l_0$	0.33 m
$m$	1.448 kg
$k_t$	$1.084 \times 10^{-5}$ kg-m
$k_d$	$1.726 \times 10^{-7}$ kg-m <sup>2</sup>
$J_x$	$1.035 \times 10^{-1}$ kg-m <sup>2</sup>
$J_y$	$1.03 \times 10^{-1}$ kg-m <sup>2</sup>
$J_z$	$1.709 \times 10^{-1}$ kg-m <sup>2</sup>

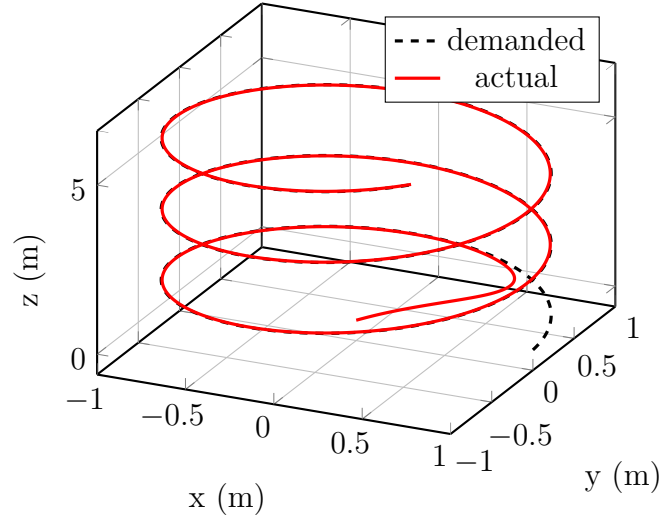


Figure 5.2: Tricopter response to spiral trajectory with SMC

and may be because the demanded trajectory in the case of SMC is different and more complex (spiral) compared to that of the 6-DOF controller.

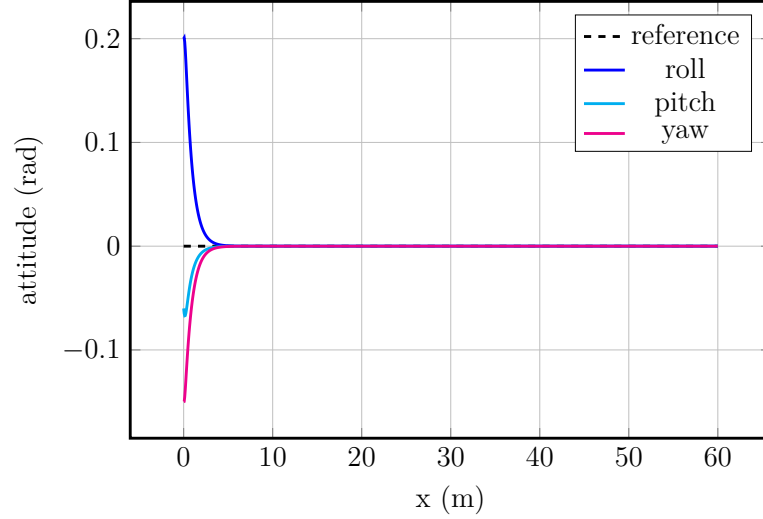


Figure 5.3: Tricopter attitude with SMC

### 5.3 System Identification of NI systems

It is often important to obtain an NI model of a physical system in order to apply NI control techniques to such a system. This section presents a continuous time method for identifying NI models by enforcing the NI property and applies it in identifying an NI model for the SMC-linearized multi-tilt tricopter UAV. This identified NI model will be used later to develop a leader-following formation scheme for networked linearized multi-tilt tricopters.

#### 5.3.1 Continuous time closed-loop system identification

In this subsection, a continuous time closed-loop identification method which guarantees that the resultant model is negative imaginary is developed. Ref. [47] gives a state-space characterization of NI systems referred to as the negative imaginary lemma given in Lemma 2.8. This result has been generalized in [73] to include poles on the imaginary axis except at the origin. Let us now recall the NI lemma as follows:

**Lemma 2.8** (NI Lemma, [47,73]). *Let  $G(s)$  be the real, rational and proper transfer function matrix of a finite-dimensional, square and causal system  $G$  having a minimal state-space realization  $\left[ \begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]$ . Then,  $G(s)$  is NI without poles at the origin if and*

only if  $\det(A) \neq 0$ ,  $D = D^\top$  and there exists a real matrix  $X = X^\top > 0$  such that

$$\begin{aligned} AX + XA^\top &\leq 0 \quad \text{and} \\ B &= -AXC^\top. \end{aligned} \tag{2.14}$$

Linearized UAV dynamics in closed-loop have been shown to have negative imaginary transfer functions [41, 55, 57]. The problem is that in order to use the result in lemma 2.8 in model identification, the identification method must be one that can characterize state-space systems. This suggests that subspace methods would be suitable. However, this poses another problem as subspace identification is typically performed in discrete time but the conditions (2.14) are in continuous time. If we transform them to discrete equivalents as done in [109] for example, the conditions (2.14) become non-convex and are no longer LMIs making the identification process more complex and less intuitive. To solve this problem, a continuous time method for identifying state-space NI models is proposed. The identification method presented is an extension of the works of [60] and [59] to characterize NI systems.

Let us consider a minimal state-space realization of a real-rational proper transfer function  $M(s)$  as  $(A, B, C, D)$  where  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{m \times n}$ ,  $D \in \mathbb{R}^{m \times m}$ , with  $u(t) \in \mathbb{R}^m$  as the input and  $y(t) \in \mathbb{R}^m$  as the output and  $x(t) \in \mathbb{R}^n$  as the states in the system. The identification process may be summarized to three main steps.

1. The bilinear relationship  $w = \frac{s-p}{s+p}$  with  $p > 0$  based on the the Laguerre filter [59] is used to obtain filtered input-output data matrices.
2. The classic subspace error-in-variables method (EIV) [110] is used to obtain the  $A$  and  $C$  matrices.
3. The  $B$  and  $D$  matrices are obtained by optimizing a least squares problem using the estimated  $A$  and  $C$  matrices with the model structure as

$$y(t|B, D) = C(qI_n - A)^{-1}Bu(t) + Du(t) \tag{5.5}$$

where  $q$  is used to represent the Laplace operator in order to differentiate from ‘ $s$ ’ used in the bilinear transformation in step 1.

The reader is referred to [59, 60] for full details of estimating the  $A$  and  $C$  matrices from closed-loop data (steps 1 and 2). In the method proposed in this work, only step

3 is modified using Lemma 2.8 to enforce the identification of an NI system. The steps and algorithm for estimating the  $A$  and  $C$  matrices (steps 1 and 2) in this study are identical to those described in [60] and are therefore, not discussed here.

Noting that for a proper system,  $D = D^T = 0$ , and let  $\hat{A}$  and  $\hat{C}$  be the estimated  $A$  and  $C$  matrices (from steps 1 and 2), (5.5) may be defined as

$$\begin{aligned} \begin{bmatrix} y(t_1) \\ y(t_2) \\ \vdots \\ y(t_\delta) \end{bmatrix} &= \begin{bmatrix} \hat{C}(qI_n - \hat{A})^{-1}u(t_1) \\ \hat{C}(qI_n - \hat{A})^{-1}u(t_2) \\ \vdots \\ \hat{C}(qI_n - \hat{A})^{-1}u(t_\delta) \end{bmatrix} \begin{bmatrix} B \end{bmatrix}, \\ \Leftrightarrow \mathbf{Y}(t) &= \mathbf{\Phi}(t)B, \\ \Leftrightarrow \mathbf{Y}(t) &= -\mathbf{\Phi}(t)\hat{A}X\hat{C}^T \text{ from (2.14) of Lemma 2.8} \end{aligned} \quad (5.6)$$

where  $X = X^T \in \mathbb{R}^{n \times n}$ ,  $t_i \forall i \in \{1, 2, \dots, \delta\}$  denotes the respective time periods of the data samples and  $\delta$  is the total no. of input-output data samples. By relating (5.6) to the normal equations for least squares [110], the constrained optimization problem is then formulated using Lemma 2.8 as

$$\min_X \sum_{t=1}^{\delta} \left\| \mathbf{\Phi}(t)^T \mathbf{Y}(t) + \mathbf{\Phi}(t)^T \mathbf{\Phi}(t) \hat{A}X\hat{C}^T \right\|^2 \quad (5.7)$$

subject to:

$$\hat{A}X + X\hat{A}^T \leq 0,$$

$$X > 0.$$

The cost function (5.7) is quadratic in  $X$  and the constraints are affine in  $X$ . Thus, the constrained optimization problem is convex and can be solved using `cvx` package in Matlab. Finally, the  $B$  matrix of the identified system is computed as  $\hat{B} = -\hat{A}X\hat{C}^T$ .

### 5.3.2 System identification of a tricopter enforcing NI property

In this subsection, the linearized multi-tilt tricopter from section 5.2.1 with six inputs  $(x_d, y_d, z_d, \phi_d, \theta_d, \psi_d)$  and their outputs (respectively) is considered as the closed-loop system to be identified. The input-output data is obtained by exciting each channel with a square wave and measuring the corresponding output signal. The continuous time

identification algorithm described in section 5.3.1 is then used to identify a model for each channel (channel by channel taking each channel as a SISO system). The Laguerre filter gain for each channel was heuristically chosen as 0.55, 0.45, 0.34, 0.35, 0.4 and 0.45 respectively, and by inspecting the singular values in the identification process, the model order was chosen as  $n = 2$  for all channels, yielding the following SISO NI transfer functions for the respective channels:

$$m_1(s) = \frac{x}{x_d} = \frac{2.3662 \times 10^{-5}s + 3.515 \times 10^{-5}}{s^2 + 0.006301s + 4.357 \times 10^{-5}}, \quad (5.8)$$

$$m_2(s) = \frac{y}{y_d} = \frac{2.659 \times 10^{-14}s + 2.714 \times 10^{-5}}{s^2 + 0.006625s + 3.682 \times 10^{-5}}, \quad (5.9)$$

$$m_3(s) = \frac{z}{z_d} = \frac{8.769 \times 10^{-14}s + 2.38 \times 10^{-5}}{s^2 + 0.006368s + 3.067 \times 10^{-5}}, \quad (5.10)$$

$$m_4(s) = \frac{\phi}{\phi_d} = \frac{6.114 \times 10^{-17}s + 4.827 \times 10^{-5}}{s^2 + 0.01231s + 5.206 \times 10^{-5}}, \quad (5.11)$$

$$m_5(s) = \frac{\theta}{\theta_d} = \frac{7.62 \times 10^{-14}s + 5.676 \times 10^{-5}}{s^2 + 0.01357s + 5.849 \times 10^{-5}}, \quad (5.12)$$

$$m_6(s) = \frac{\psi}{\psi_d} = \frac{1.419 \times 10^{-14}s + 1.49 \times 10^{-4}}{s^2 + 0.003192s + 1.325 \times 10^{-4}}. \quad (5.13)$$

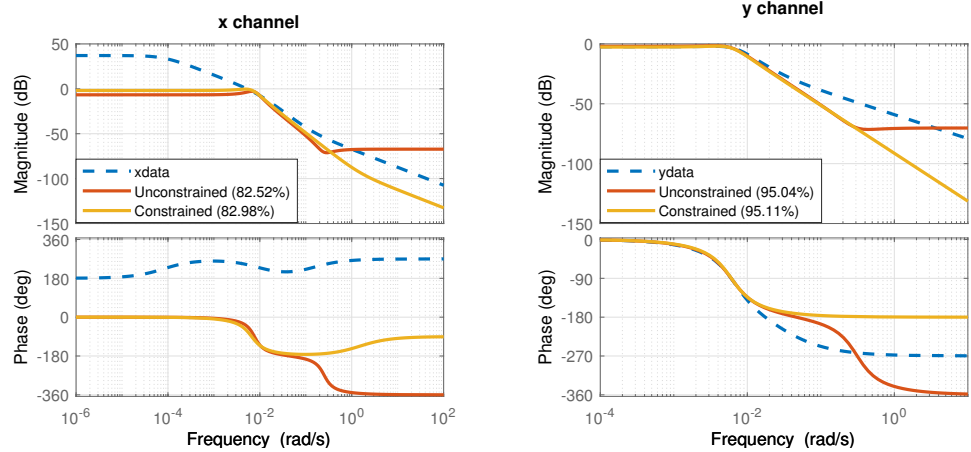


Figure 5.4: Frequency response comparison:  $x$  and  $y$  channels

Equations (5.8) to (5.13) together represent the SMC-linearized dynamics of the multi-tilt tricopter in closed-loop with six inputs and six outputs. Figures 5.4 to 5.6 show the frequency response validation of the identified models together with a comparison of these models with the classic subspace identification algorithm [110,111]. In Figures 5.4 to 5.6, *unconstrained* refers to the classic subspace algorithm [111] while *constrained* refers to the proposed continuous time algorithm which solves the constrained optimization problem (5.7) and guarantees that the identified model is NI.



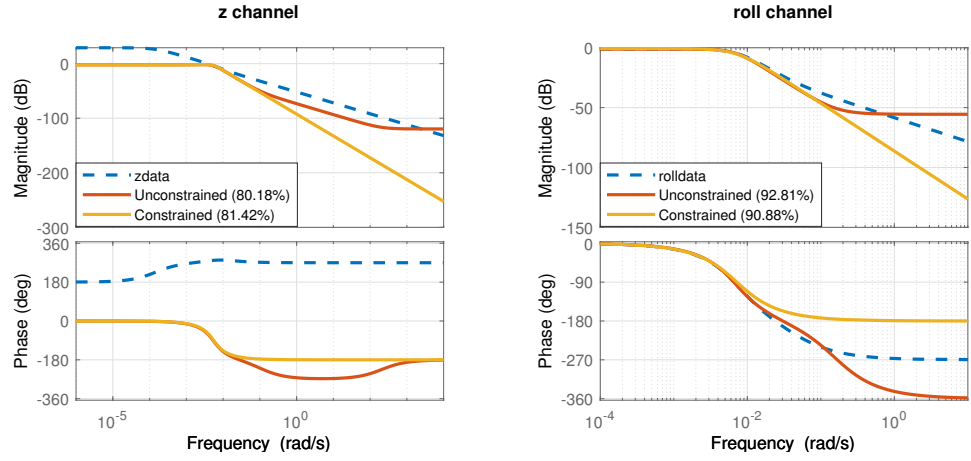
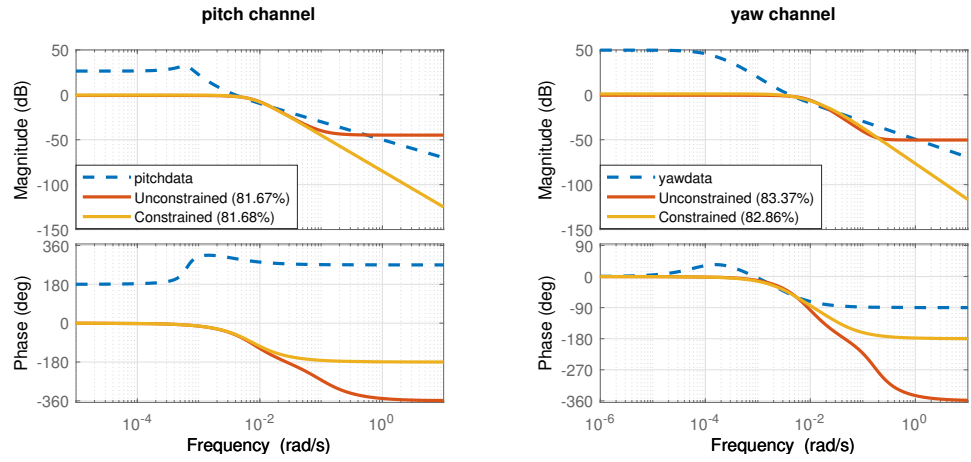

 Figure 5.5: Frequency response comparison:  $z$  and roll channels


Figure 5.6: Frequency response comparison: pitch and yaw channels

It is evident from Figures 5.4 to 5.6 that the identified models for each channel using the proposed algorithm have phase in the range  $[-\pi, 0]$  which is required for a negative imaginary system. Furthermore, it can also be observed from Figures 5.4–5.6 that for all channels except roll and yaw, the proposed algorithm (constrained) yields better fits with the validation data compared to the classic subspace algorithm (unconstrained).

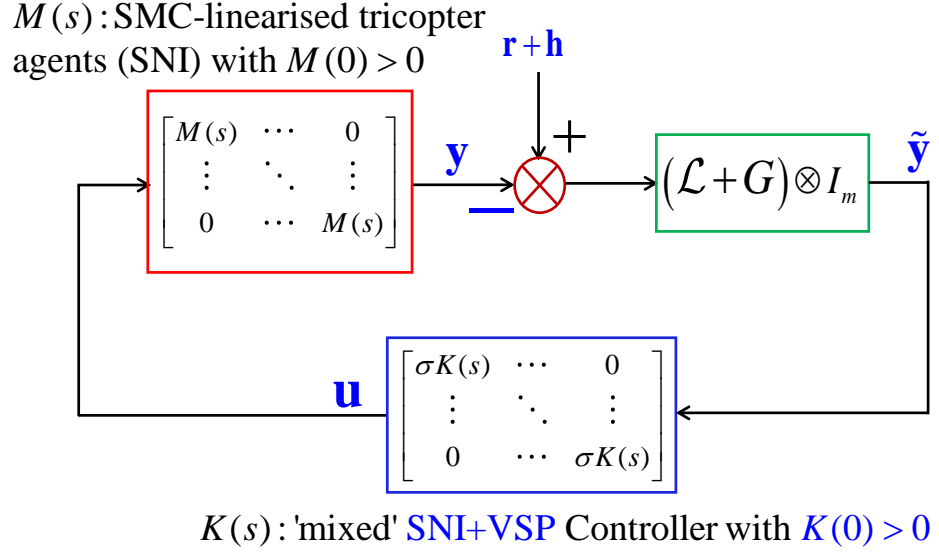


Figure 5.7: A formation control scheme for a group of networked SMC-linearized tricopter agents (being SNI) involving a decoupled ‘mixed’ SNI+VSP controller  $K(s)$  with  $K(0) > 0$ .

## 5.4 Formation control of multi-tilt tricopters

This section lays down the foundational results which underpin the main contributions of this chapter that will be developed in the subsequent sections. Here, it will be established that a network of SMC-linearized agents can be made closed-loop stable via a distributed ‘mixed’ SNI+VSP controller depending only on the sign definiteness of the DC-gain matrix of the controller transfer function. This result will be invoked later to develop a leader-following formation control scheme (shown in Figure 5.7) for networked SMC-linearized multi-tilt tricopter agents.

### 5.4.1 Closed-loop stability of networked NI/SNI system with ‘mixed’ SNI+VSP controller

In this subsection, it will first be established (in Lemma 5.1) that a network of SMC-linearized multi-tilt tricopter agents  $M(s) = \text{diag}\{m_1(s), m_2(s), \dots, m_m(s)\}$  as derived in (5.8)–(5.13), being stable NI or SNI with  $M(0) > 0$ , connected via an undirected graph, can be stabilized in a negative feedback loop shown in Figure 5.7 by a distributed ‘mixed’ SNI+VSP controller  $K(s)$  having  $K(0) > 0$ . The terminology ‘mixed’ SNI+VSP controller signifies that the elements of  $K(s)$  exhibit either purely SNI

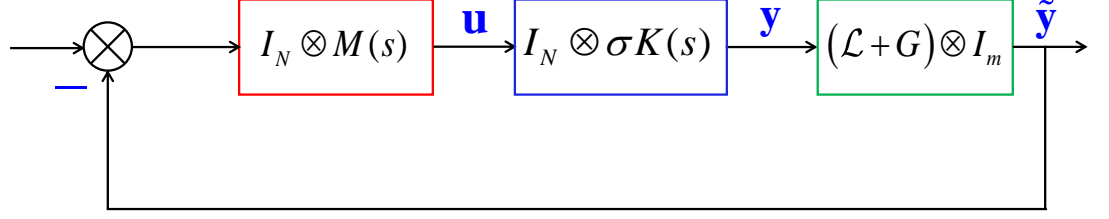


Figure 5.8: Closed-loop stability of a negative feedback interconnection containing a networked ‘mixed’ SNI+VSP system  $K(s)$  cascaded with a decentralised SNI system  $M(s)$ .

property, or purely VSP property, or a *mixture* of SNI and VSP properties. In other words, some of the elements of  $K(s)$  may be SNI, some of the elements may be VSP while the rest satisfy both SNI and VSP properties. Theorem 5.1 is the main contribution of this section which proves that a group of networked multi-tilt tricopter agents achieves a predefined time-invariant or time-varying formation under the application of a distributed ‘mixed’ SNI+VSP controller  $K(s)$ , described before, when  $K(0) > 0$ . Lemma 5.1 is an essential technical pre-requisite result, which will be invoked for proving Theorem 5.1. The proof of Lemma 5.1 significantly relies on the Eigenvalue loci theory [71, 72] and has been done taking inspiration from [70] and [42].

**Lemma 5.1.** *Consider a network of  $N$  identical, decoupled stable NI/SNI systems  $M(s) \in \mathcal{RH}_\infty^{m \times m}$  with  $M(0) > 0$ . Let the graph  $\mathcal{G}$  associated with the network satisfy Assumption 2.1. Then, there exists a finite range of  $\sigma \in (0, \sigma^*]$  for which the negative feedback interconnection of  $[(\mathcal{L} + G) \otimes \sigma K(s)]$  and  $M(s)$  shown in Figure 5.8 remains asymptotically stable where  $K(s) \in \mathcal{RH}_\infty^{m \times m}$  is a decoupled, ‘mixed’ SNI+VSP system satisfying  $K(0) > 0$ .*

**Proof.** In this proof, the notation  $\rho_i(s)$  is used to represent the eigenvalue loci of the networked loop transfer function matrix  $[(\mathcal{L} + G) \otimes K(s)M(s)]$ . For convenience, let us define the following two sets of the complex variable  $s$  along the  $s$ -plane  $D$ -contour shown in Figure 5.9b

$$\begin{aligned} \Omega_{\pm j} &= \{s \mid s = j\omega, \omega \in (-\infty, \infty)\}, \\ \Omega_R &= \{s \mid s = Re^{j\theta}, R \in \mathbb{R}_{>0}, R \rightarrow +\infty, -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}\}. \end{aligned}$$

The negative feedback interconnection of  $M(s)$  and  $[(\mathcal{L} + G) \otimes K(s)]$ , as shown in Figure 5.8, remains asymptotically stable if none of the eigenvalue loci  $\rho_i(j\omega)$  encircles

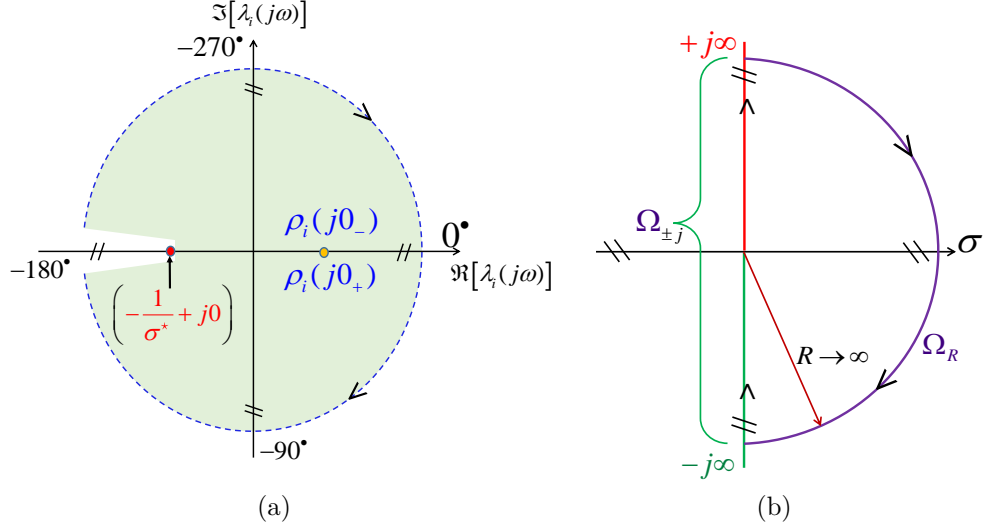


Figure 5.9: (a) All the eigenvalue loci  $\rho_i(j\omega)$  of  $(\mathcal{L} + G) \otimes K(s)M(s)$  remain confined within the Green coloured region  $\forall \omega \in \mathbb{R} \cup \{\infty\}$  when  $M(s)$  is SNI with  $M(0) > 0$  and  $K(s)$  is ‘mixed’ SNI+VSP with  $K(0) > 0$ ; (b) Nyquist  $\mathcal{D}$ -contour in the  $s$ -plane.

the critical point  $(-\frac{1}{\sigma} + j0)$  for any  $\sigma \in (0, \sigma^*]$  via Theorem 2.2. Let us now establish via the following two parts (Parts I and II) that all the eigenvalue loci  $\rho_i(s)$  remain confined within the Green coloured region portrayed in Figure 5.9a. Before starting the proof, it is worth noting that both  $M(s) \in \mathcal{RH}_\infty^{m \times m}$  and  $K(s) \in \mathcal{RH}_\infty^{m \times m}$  have decoupled structures, that is,  $M(s) = \text{diag}\{m_1(s), m_2(s), \dots, m_m(s)\}$  and  $K(s) = \text{diag}\{k_1(s), k_2(s), \dots, k_m(s)\}$ . Since  $M(s)$  is stable NI and  $K(s)$  is ‘mixed’ SNI+VSP respectively,  $\angle m_i(j\omega) \in [-\pi, 0]$  and  $\angle k_i(j\omega) \in (-\pi, \frac{\pi}{2})$  for all  $\omega \in (0, \infty)$  and for all  $i \in \{1, 2, \dots, m\}$ . Also,  $\angle m_i(0) = 0$  and  $\angle k_i(0) = 0$  for all  $i$  since  $M(0) = M(0)^\top > 0$  and  $K(0) = K(0)^\top > 0$  via supposition and due to satisfying the stable NI/SNI properties. It is also assumed that  $[M(s) - M^\sim(s)]$  has full normal rank.

Part I: When  $s \in \Omega_{\pm j}$  [i.e. when  $\omega \in (-\infty, \infty)$ ]

Let  $\lambda_i [(\mathcal{L} + G) \otimes K(j\omega)M(j\omega)] = \lambda_i [(\mathcal{L} + G)] |k_i(j\omega)| |m_i(j\omega)| e^{j(\phi_i + \psi_i)}$  at each  $\omega \in (0, \infty)$  and for all  $i \in \{1, 2, \dots, Nm\}$ . Since  $M(s)$  is stable NI and  $K(s)$  is ‘mixed’ SNI+VSP,  $\psi_i(\omega) \in [-\pi, 0]$  and  $\phi_i(\omega) \in (-\pi, \frac{\pi}{2})$  for all  $\omega \in (0, \infty)$  and hence,  $\angle \rho_i(j\omega) = (\phi_i(\omega) + \psi_i(\omega)) \in [-2\pi, 0]$  for all  $\omega \in (0, \infty)$  and for all  $i$ . Similarly, for all  $\omega \in (-\infty, 0)$ ,  $\angle \rho_i(j\omega) \in [-2\pi, 0]$ . At  $\omega = 0$ , we have  $\lambda_i [(\mathcal{L} + G) \otimes K(0)M(0)] = \lambda_i [(\mathcal{L} + G)] |k_i(0)| |m_i(0)| \angle 0$  as  $K(0) > 0$  and  $M(0) > 0$ . Therefore, the zero-frequency points  $\rho_i(j0_-)$  and  $\rho_i(j0_+)$  of all the eigenvalue loci  $\rho_i(s)$  lie on the positive real axis of the eigenvalue loci plane and they coincide as both  $K(s)$  and  $M(s)$  do not have any

pole(s) at the origin. Thus, when  $s \in \Omega_{\pm j}$ , the angle contribution  $\angle \rho_i(j\omega)$  of each of the eigenvalue loci belongs to the range  $[-2\pi, 0] \forall \omega \in \mathbb{R}$ . Most importantly, there is no infinite crossover on the negative and positive real axis as neither of  $K(s)$  and  $M(s)$  contains a pole at the origin. Furthermore, the eigenvalue loci  $\rho_i(s)$  may intersect the negative real axis one or multiple times at finite distances since the intercept, given by  $\lambda_i [(\mathcal{L} + G)] |k_i(j\omega)| |m_i(j\omega)|$ , remains finite at all  $\omega \in \mathbb{R}$ . Since the eigenvalue loci  $\rho_i(s)$  intersect the negative real axis at finite distances, there always exists a finite range  $(0, \sigma^*]$  of the parameter  $\sigma$  for which the critical point  $(-\frac{1}{\sigma} + j0)$  is never encircled by any of the eigenvalue loci. The fact has been graphically demonstrated in Figure 5.9a. It shows that all  $\rho_i(j\omega)$  stay within the Green coloured region of the eigenvalue loci plane such that the critical point  $(-\frac{1}{\sigma} + j0)$  is never encircled for any  $\sigma \in (0, \sigma^*]$ .

#### Part II: When $s \in \Omega_R$

Similar to the zero-frequency points  $\rho_i(j0_-)$  and  $\rho_i(j0_+)$ , the infinite frequency points  $\rho_i(+j\infty)$  and  $\rho_i(-j\infty)$  can be expressed as  $\lambda_i [(\mathcal{L} + G) \otimes K(\infty)M(\infty)] = \lambda_i [(\mathcal{L} + G)] |k_i(\infty)| |m_i(\infty)| \angle (\phi_i(\infty) + \psi_i(\infty))$  for all  $i \in \{1, 2, \dots, Nm\}$ . Since the eigenvalues of  $K(\infty)M(\infty)$  are always real numbers (positive/negative/zero),  $\lambda_i [(\mathcal{L} + G)] > 0$  for all  $i$  and neither of  $K(s)$  and  $M(s)$  contains a pole at the origin, the infinite frequency points  $\rho_i(+j\infty)$  and  $\rho_i(-j\infty)$  coincide and lie either at the origin or on the real axis at finite distances from the origin. Combining all these arguments, we can conclude that there always exists a finite range  $(0, \sigma^*]$  of the gain parameter  $\sigma$  for which none of the eigenvalue loci  $\rho_i(j\omega)$  encircles the critical point  $(-\frac{1}{\sigma} + j0)$  for the entire frequency range  $\omega \in \mathbb{R} \cup \{\infty\}$  [this has been demonstrated in Figure 5.9a, which shows that all  $\rho_i(s)$  remain within the Green coloured region and the worst-case critical point  $(-\frac{1}{\sigma^*} + j0)$  also lies outside the Green coloured region].

Parts I and II together prove that all the eigenvalue loci  $\rho_i(s)$  of the loop transfer function  $[(L + G) \otimes K(s)M(s)]$  remain within the Green coloured region shown in Figure 5.9a and hence, none of the eigenvalue loci  $\rho_i(s)$  encircles the critical point  $(-\frac{1}{\sigma} + j0)$  for any  $\sigma \in (0, \sigma^*]$ . This proves asymptotic stability of the negative feedback closed-loop system shown in Figure 5.8 exploiting Theorem 2.2. This completes the proof. It is worth mentioning that the same proof holds for the cases when  $M(s) \in \mathcal{RH}_{\infty}^{m \times m}$  belongs to the SNI class for which the full normal rank condition [i.e.  $M(s) - M^{\sim}(s)$  has full normal rank] is inherently satisfied. ■

### 5.4.2 Formation control of networked multi-tilt tricopters using ‘mixed’ SNI+VSP controller

This subsection presents the key contribution of this chapter. NI and passivity theories are used to design a simple leader-following formation-seeking scheme as depicted in Figure 5.7 for a class of UAVs that can be modelled as (or transformed into) a network of linearized dynamics. Before discussing the scheme, let us declare the set of admissible reference input signals  $r(t)$  (generated by the leader or the root node) to be followed by the agents.

**Assumption 5.1.** *Let  $\mathbf{r}_0(t) = [r_1(t) \ r_2(t), \dots, r_m(t)]^\top \in \mathbb{R}^m \ \forall t \geq 0$  be the given tracking reference signal from the root node where  $r_i$  is an  $\mathcal{L}_\infty$ -bounded signal for all  $i$ , so that  $\mathbf{r} = [\mathbf{r}_0^\top \ \mathbf{r}_0^\top, \dots, \mathbf{r}_0^\top]^\top \in \mathbb{R}^{mN}$ .*

**Theorem 5.1.** *Consider a network of  $N$  identical SMC-linearised, multi-tilt, tricopter agents  $M(s) = \text{diag}\{m_1(s), m_2(s), \dots, m_m(s)\} \in \mathcal{RH}_\infty^{m \times m}$  with  $M(0) > 0$ , as derived in (5.8)–(5.13), connected via the topology  $\mathcal{G}$  that satisfies Assumption 2.1. The set of admissible formation reference inputs  $\mathbf{r}(t)$  satisfies Assumption 5.1 and let  $\mathbf{h} = [\mathbf{h}_1^\top \ \mathbf{h}_2^\top, \dots, \mathbf{h}_N^\top]^\top \in \mathbb{R}^{mN}$  be the desired formation configuration vector. Let  $K(s) \in \mathcal{RH}_\infty^{m \times m}$  be a decoupled, ‘mixed’ SNI+VSP system satisfying  $K(0) > 0$  and there exists a finite  $\sigma^* > 0$ . Then, the network of tricopter agents achieves the desired formation with respect to  $\mathbf{r}(t)$  and  $\mathbf{h}$  by the following distributed dynamic output feedback control law (shown in Figure 5.7)*

$$\mathbf{u}_i = \sigma K(s) \sum_{j=1}^N a_{ij} \left( (\mathbf{y}_j - \mathbf{h}_j) - (\mathbf{y}_i - \mathbf{h}_i) \right) + g_i (\mathbf{r}_0 + \mathbf{h}_i - \mathbf{y}_i) \quad (5.15)$$

$\forall i \in \{1, 2, \dots, N\}$  and for any  $\sigma \in (0, \sigma^*]$  with  $g_i$  as the pinning gain.

**Proof:** Let us first note that the SMC-linearised model  $M(s) \in \mathcal{RH}_\infty^{6 \times 6}$  of the multi-tilt tricopter considered in this paper satisfies SNI property with  $M(0) > 0$ . The proposed formation control scheme for networked multi-tilt tricopter agents is shown in Figure 5.7. An equivalent block diagram of Figure 5.7 has been drawn in Figure 5.10 to assist the proof of Theorem 5.1. Let us denote  $\mathcal{L}_G = \mathcal{L} + G$ . The proof builds on Lemma 5.1, which establishes the internal asymptotic stability of a negative feedback interconnection containing a networked stable NI/SNI system  $M(s)$  and a decoupled

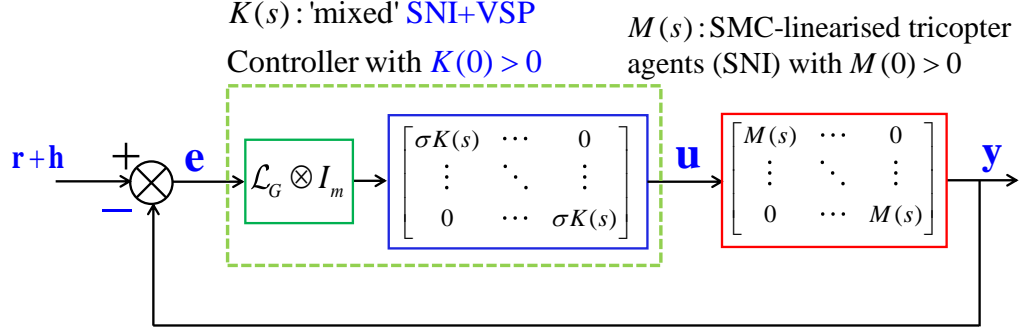


Figure 5.10: An equivalent block diagram of the formation control scheme shown in Figure 5.7 for SMC-linearised networked tricopter agents. Note  $\mathcal{L}_G = (\mathcal{L} + G)$ .

‘mixed’ SNI+VSP system  $K(s)$  exploiting the eigenvalue loci technique. We may recall here that the interaction topology ( $\mathcal{G}$ ) among the agents satisfies Assumption 2.1 and the admissible set of formation reference signals (generated by a command generator, which can be treated as an exo-system) obeys Assumption 5.1.

In Figure 5.10, the Green dotted box represents the distributed ‘mixed’ SNI+VSP controller  $\mathcal{L}_G \otimes \sigma K(s)$ . For the purpose of the proof, the network part (i.e.  $\mathcal{L}_G = \mathcal{L} + G$ ) has been decoupled from the plant and attached with the controller block. Now, the negative feedback interconnection of the SMC-linearised multi-tilt tricopter system  $\text{diag}\{M(s), M(s), \dots, M(s)\}$  and the networked controller  $\mathcal{L}_G \otimes \sigma K(s)$  in Figure 5.10 is internally asymptotically stable for a finite range of the gain factor  $\sigma \in (0, \sigma^*]$  via Lemma 5.1 as the block diagram shown in Figure 5.8 is equivalent to Figure 5.10. The asymptotic stability of the networked loop ensures that trajectory tracking error will asymptotically decay to zero, that is,  $\lim_{t \rightarrow \infty} \mathbf{e}(t) = 0$  or  $\lim_{t \rightarrow \infty} \mathbf{r}(t) + \mathbf{h}(t) - \mathbf{y}(t) = 0$ . This part readily follows from Theorem 1 of [112]. Hence, we can conclude that the group of SMC-linearised multi-tilt tricopter agents will achieve the desired formation specified by  $\mathbf{r}$  and  $\mathbf{h}$  under the influence of the distributed ‘mixed’ SNI+VSP controller  $K(s) \in \mathcal{RH}_\infty^{6 \times 6}$  with  $K(0) > 0$  following the scheme shown in Figure 5.7 [equivalently Figure 5.10]. ■

**Remark 5.1.** The negative feedback consensus-seeking scheme developed for SMC-linearized tricopter agents may be easily modified to cater to single and double integrator agents. In that respect, the results presented in [42, 80] where tricopters were feedback-linearized to single/double integrator systems can be captured by the scheme presented in this chapter. However, it has been investigated that in the case of single

integrator agents, if a negative feedback consensus scheme is used, then the requirement of an SNI controller can be relaxed to a stable NI (as opposed to an SNI) controller. This reduces the conservatism of the proposed consensus scheme and hence, is worth detailed analysis.

## 5.5 Case study and simulation results

This section presents the formation control design for a network of six multi-tilt tricopter agents using the results developed in section 5.4. These six agents have identical NI dynamics based on the SMC-linearized model identified in subsection 5.3.2. Matlab simulation results are presented to show the effectiveness of the proposed scheme.

### 5.5.1 Formation control of a group of six tricopters

For this case study, consider a set of six multi-tilt tricopter UAVs escorting a leader or target and whose goal is to track a predefined formation. Each of the six agents has identical dynamics which can be described by the linear state-space model

$$\dot{x}_i = Ax_i + Bu_i \quad \text{and} \quad y_i = Cx_i \quad \forall i \in \{1, 2, \dots, 6\},$$

where  $x_i = [\dot{x}_i \ x_i \ \dot{y}_i \ y_i \ \dot{z}_i \ z_i \ \dot{\phi}_i \ \phi_i \ \dot{\theta}_i \ \theta_i \ \dot{\psi}_i \ \psi_i]^\top$ ,  $u_i = [x_{di} \ y_{di} \ z_{di} \ \phi_{di} \ \theta_{di} \ \psi_{di}]^\top$ ,  $y_i = [x_i \ y_i \ z_i \ \phi_i \ \theta_i \ \psi_i]^\top$  and the matrices  $A, B, C$  are given as

$$A = \begin{bmatrix} -63 & -56 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 78 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -66 & -47 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 78 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -64 & -39 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 78 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -123 & -67 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 78 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -136 & -75 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 78 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -32 & -85 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 156 & 0 \end{bmatrix} \times 10^{-4},$$



$$B = \begin{bmatrix} 625 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 625 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 625 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 625 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 625 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1250 \end{bmatrix} \times 10^{-4} \text{ and}$$

$$C = \begin{bmatrix} 4 & 720 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 556 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 487 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 989 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1162 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 763 \end{bmatrix} \times 10^{-4}.$$

The above model represents the SMC-linearized multi-tilt tricopter system  $M(s) = \text{diag}\{m_j(s)\} \in \mathcal{RH}_\infty^{6 \times 6} \forall j \in \{1, 2, \dots, 6\}$  using (5.8) to (5.13), with  $m$  inputs and outputs, which was identified in subsection 5.3.2, and satisfies the NI property. It can also be easily verified that  $(A, B, C)$  is stabilizable and detectable. The undirected interaction topology among the six vehicles is given in Figure 5.11, where the leader agent or target (labelled 0) provides the formation reference signal.

It follows from Figure 5.11 that the Laplacian of the network is

$$\mathcal{L} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 & 0 \\ -1 & -1 & 4 & 0 & -1 & -1 \\ 0 & -1 & 0 & 2 & -1 & 0 \\ 0 & -1 & -1 & -1 & 4 & -1 \\ 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix}, \quad (5.16)$$

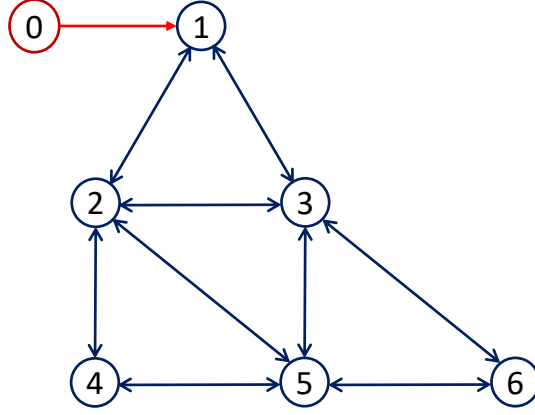


Figure 5.11: Undirected network interaction topology.

and since only agent 1 is connected to the leader, an edge  $(0,1)$  exists between them with a pinning gain  $g_1 = 1$  while  $g_i = 0$  for  $i \in \{2, \dots, 6\}$ .

A high-gain SNI controller is chosen as  $k_1(s) = \frac{10^7}{(s+\rho_1)}$  to stabilize the  $x$  position, and a set of high-gain VSP controllers of the form  $k_j(s) = \frac{8 \times 10^7 (s+2)}{(s+\rho_j)} \forall j \in \{2, 3, \dots, 6\}$  are chosen for the rest of the channels with  $\rho = [\rho_1, \rho_2, \dots, \rho_6] = [15, 100, 80, 90, 95, 49]$ . These high gains were necessary to satisfy the control requirements. Consequently, the ‘mixed’ SNI+VSP controller is given as  $K(s) = \text{diag}\{k_j\} \forall j \in \{1, 2, \dots, 6\}$  with  $\sigma = 1$ , and it is trivial to show that  $K(0) > 0$  as required. As depicted in Figure 5.10, the group formation reference which achieves the desired formation is given as  $(\mathbf{r} + \mathbf{h}) \in \mathbb{R}^{36}$  where  $\mathbf{r} = [\mathbf{r}_0^\top \mathbf{r}_0^\top, \dots, \mathbf{r}_0^\top]^\top \in \mathbb{R}^{36}$  is the formation reference and  $\mathbf{h} = [\mathbf{h}_1^\top \mathbf{h}_2^\top, \dots, \mathbf{h}_6^\top]^\top \in \mathbb{R}^{36}$  is the formation configuration. The reference for the leader or root node is selected as  $\mathbf{r}_0 = [4 \ 3 \ 2 \ 0 \ 0 \ 0]^\top \forall t < 10\text{s}$  and  $\mathbf{r}_0 = [-1 \ 5 \ 3.2 \ 0 \ 0 \ 0]^\top \forall t \geq 10\text{s}$ .

The formation configuration for the followers was chosen as

$$\mathbf{h}_i = \begin{bmatrix} \gamma \cos\left(\frac{(i+1)\pi}{3}\right) \\ \gamma \sin\left(\frac{(i+1)\pi}{3}\right) \\ 0 \\ \gamma \sin\left(\frac{(i+1)\pi}{3}\right) \\ \gamma \cos\left(\frac{(i+1)\pi}{3}\right) \\ 0 \end{bmatrix} \forall t < 10\text{s}$$

and

$$\mathbf{h}_i = \begin{bmatrix} \beta_i \cos\left(\frac{(2i+1)\pi}{6}\right) \\ \beta_i \sin\left(\frac{(2i+1)\pi}{6}\right) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \forall t \geq 10s, \forall i \in \{1, 2, \dots, 6\},$$

where  $\gamma = 2.0$  m is the radius of the formation,  $\beta_i = \gamma \forall i \in \{1, 3, 5\}$  and  $\beta_i = \gamma/2 \forall i \in \{2, 4, 6\}$ . Initially with  $t < 10s$ , the follower agents track a diamond formation bordering the leader agent, while for  $t \geq 10s$ , the follower agents track a triangle formation with respect to the formation configuration  $\mathbf{h}_i$ , taking into account a change in the leader reference  $\mathbf{r}$  at  $t = 10s$ . Figure 5.12 shows that the six multi-tilt tricopter agents achieve consensus and leader-following formation tracking as  $t \rightarrow \infty$ . Note that if  $\mathbf{h} = 0$ , the entire scheme in Figure 5.10 reduces to a consensus problem. It therefore follows that the SNI+VSP controller inherently achieves consensus tracking.

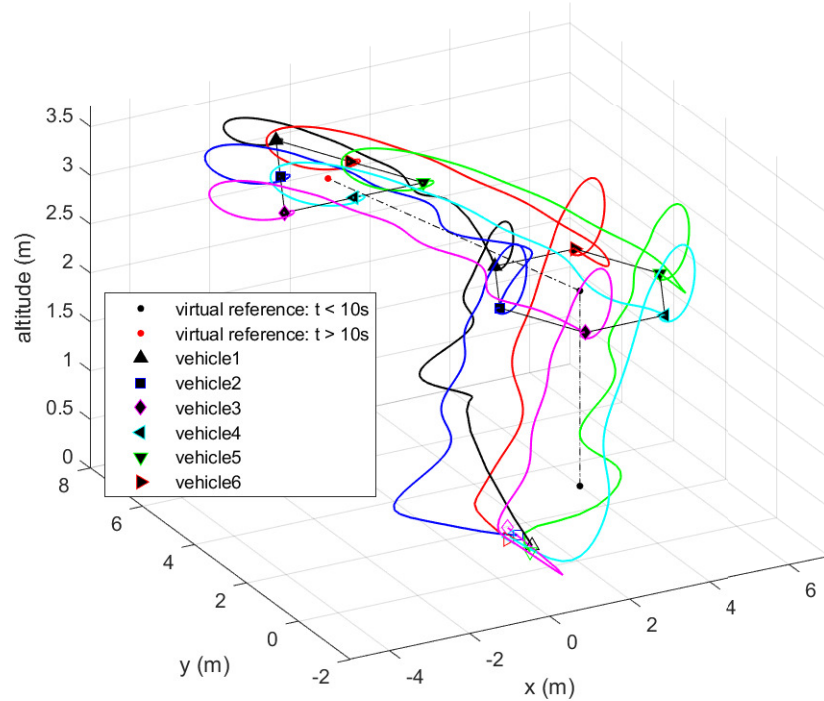
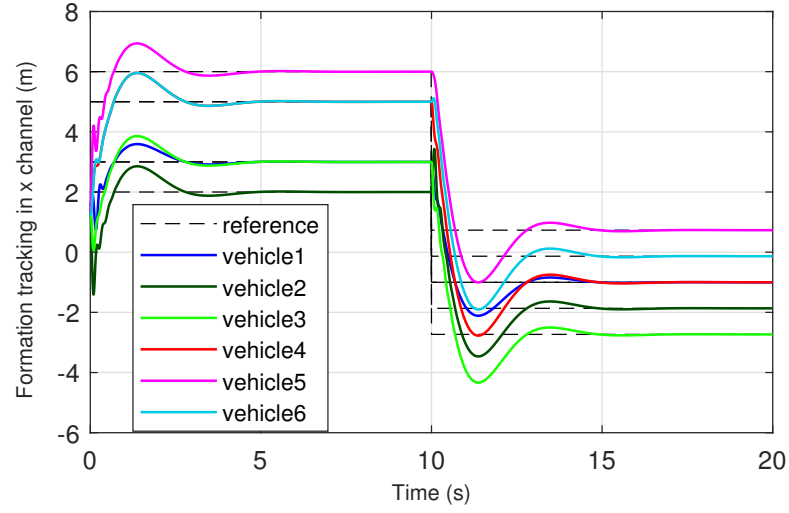
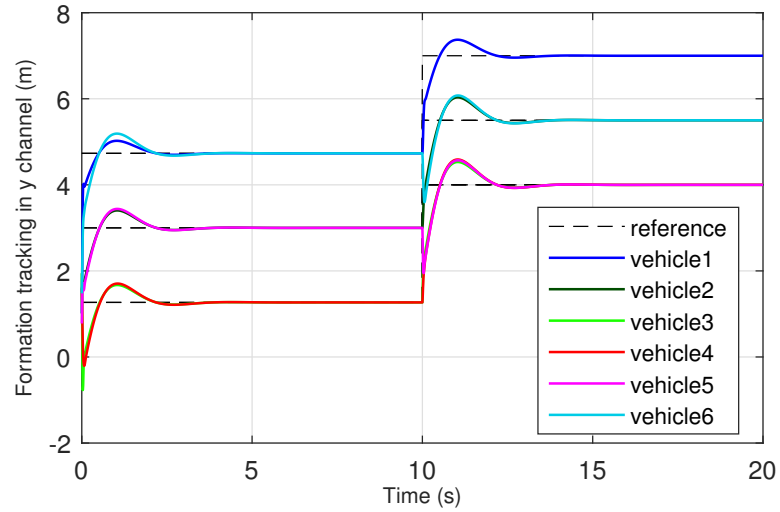
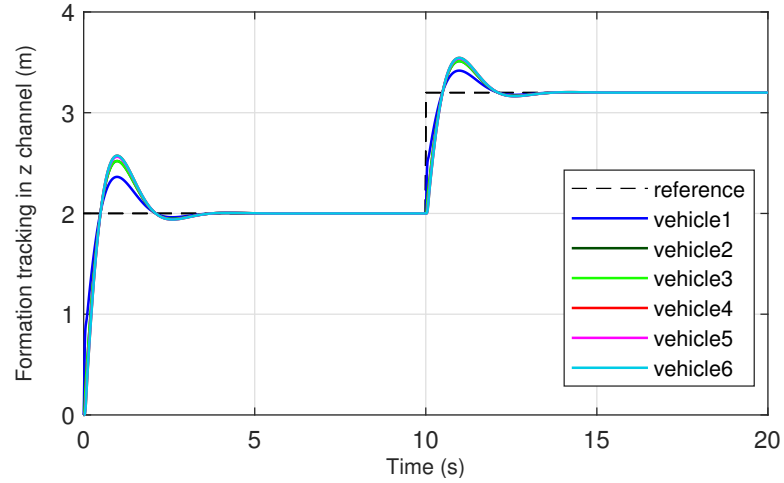


Figure 5.12: Group formation of six tricopter agents with SNI+VSP controllers with formation configuration switched from diamond to triangle after 10 seconds.


 Figure 5.13:  $x$  position responses.

When the leader reference and demanded group formation changes at  $t = 10$ s, the SNI+VSP control scheme maintains stability of the entire network and ensures that the six multi-tilt tricopter agents track the newly demanded triangle formation. Figures 5.13 to 5.15 show the responses of the positions  $x_i, y_i, z_i \forall i \in \{1, 2, \dots, 6\}$  to the demanded references.


 Figure 5.14:  $y$  position responses.

Figure 5.15: altitude  $z$  responses.

It is clear that for the six follower agents, the actual positions track the demanded position with respect to the leader within 5 seconds both before and after the time period  $t \geq 0$  which corresponds to a change in the demanded group formation. Similarly, Figures 5.16 to 5.18 show that the attitudes of the six multi-tilt tricopter agents track the target attitude.

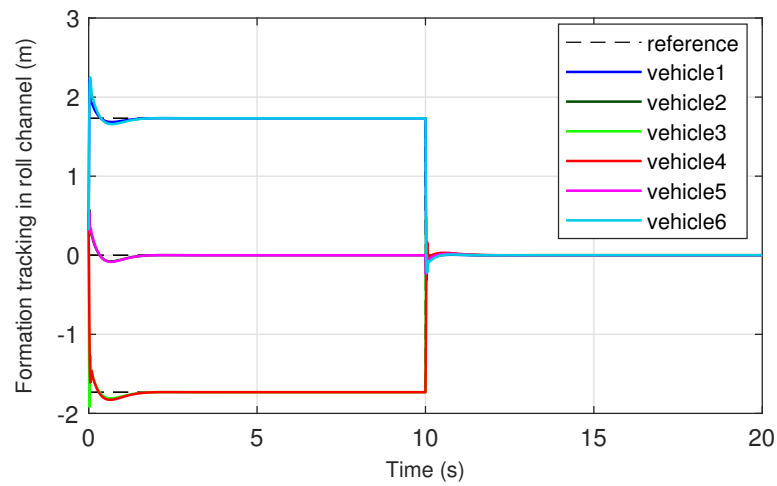


Figure 5.16: roll attitude responses.

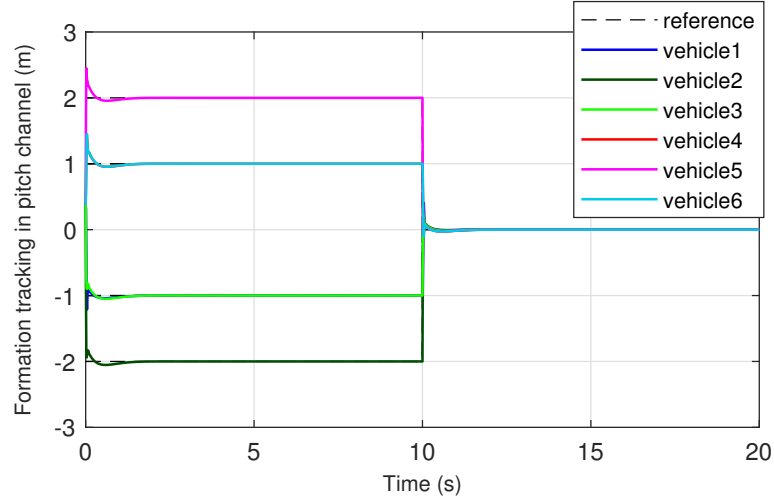


Figure 5.17: pitch attitude responses.

Considering Figures 5.16 to 5.18, it is also worth noting that for  $t < 10$ s, the motion of the multi-tilt tricopter agents occurs with a change in attitude, implying simultaneous position and attitude control, which is an advantage of the tricopter under consideration due to its airframe configuration.

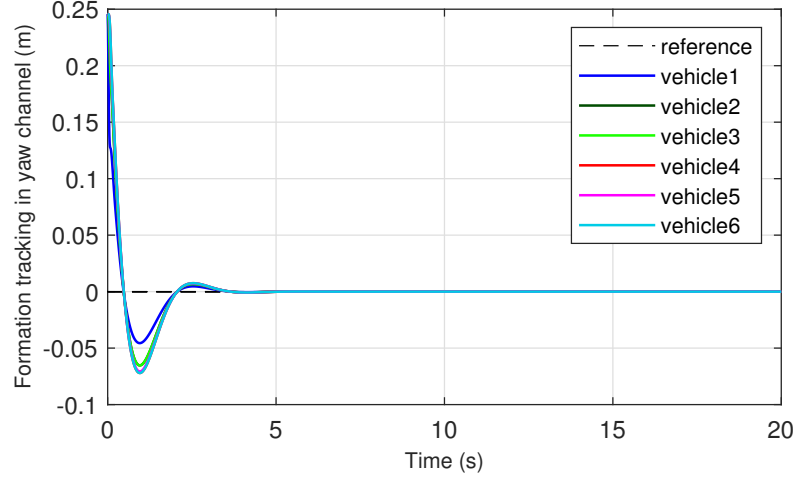


Figure 5.18: yaw attitude responses.

For  $t > 10$ s, the case where the tricopter agents translate to track the leader or target without a change in attitude is evident, indicating the ability of the tricopter under consideration to achieve simultaneous independent attitude and trajectory control, extending the limits of the classic tricopter [75] and other commonly used multicopters such as the quadcopter.

## 5.6 Summary

In this chapter, the NI theory has been exploited to solve a formation tracking problem for a network of linearized multi-tilt tricopter agents using a distributed SNI+VSP control protocol. A continuous-time system identification method has also been developed to obtain the model of a physical system in closed-loop, while guaranteeing that the model is negative imaginary. This algorithm has been applied in the identification of the closed-loop dynamics of the multi-tilt tricopter UAV. Sliding mode control laws are then developed to linearize the highly-coupled and nonlinear model. As SMC is a nonlinear technique, it preserves robustness as it does not invert all nonlinear dynamics, unlike the classic Jacobi linearization (using the pseudo inverse of the Jacobian matrix) [113] for example. Using the developed identification method, the SMC-linearized inner-loop is then identified as a negative imaginary system with six inputs and outputs corresponding to the three cartesian positions and three attitudes. A distributed formation tracking control protocol was then developed for a group of multi-tilt tricopter agents connected via an undirected graph, which guarantees that the target position and attitude of each UAV can be achieved independently. The method relies on the eigenvalue loci technique instead of the Lyapunov stability approach commonly used in the cooperative control literature. The advantage of this method is that it reduces the complexity of the results and makes the controller selection process more manageable. This is because the proposed scheme depends only on the DC gain of the SNI controller. Furthermore, by exploring the eigenvalue loci of NI/SNI systems instead of using the traditional NI/SNI stability result, the proposed results are applicable to both positive and negative feedback interconnections thereby reducing their conservatism as compared to conventional NI/SNI results which do not apply to negative feedback interconnections. The simulation results have shown the efficiency of the proposed methods with the design of a formation control scheme for a group of six NI tricopter agents. In the future, obstacle avoidance and collision will be taken into consideration.

# Chapter 6

## Concluding Remarks

In this chapter the main contributions of this thesis are summarised and possible directions for future research are outlined. The aim of this thesis is to develop advanced control techniques for tricopter UAVs going from platform design and development to operation and control. This aim has been achieved with the scratch build of two tricopter platforms together with design of varied control techniques some of which have been validated with experiments.

### 6.1 Contributions

The main contributions of this thesis to the state-of-the-art are summarized as follows:

- Design and construction of servo tilt-mechanism which extends the tilt angle range of the rotors in a multi-tilt tricopter.
- Development of control allocation scheme which allocates the actuator signals via a non-square mixer matrix due to higher number of forces and drag torques acting on the single-tilt tricopter than actuators.
- Demonstration of the capability of the multi-tilt airframe to translate in the lateral and longitudinal directions without changing its attitude by feedback linearization of the nonlinear translational equations to obtain control forces in  $x$  and  $y$  axes which are directly applied to the UAV.
- Development of a direct (manual) actuation method implemented via the control allocation of the PX4 setup, to control the lateral and longitudinal motions of



the multi-tilt tricopter while overcoming the limitation of regular Radio transmitters in supporting more than 4 DOF control. The developed method allows a regular transmitter to be used in controlling both the attitude and positions independently.

- Development of a closed-loop system identification algorithm that guarantees that the resultant model is NI. The algorithm presented exploits the classic subspace method but in continuous time (rather than discrete) by use of the Laguerre filter in the identification process.
- Development of closed-loop stability result for a network of NI/SNI systems which are controlled by ‘mixed’ Strictly Negative Imaginary (SNI) plus Very Strictly Passive (VSP) controllers in a negative feedback interconnection.
- Design of an output feedback distributed SNI+VSP control law (making use of the closed-loop stability result for networked NI/SNI systems) for achieving robust cooperative control combining both leader-following consensus and formation control. Rather than the Lyapunov-based approach, the characteristics of the eigenvalue loci of networked NI and SNI systems is used to prove the convergence of the states.

The other technical contributions of this thesis are summarized as follows:

- Development of a single-tilt tricopter using low-cost materials and open-source software (PX4) with optical flow included for GPS-denied environments. The dynamic model is derived, an experiment to obtain actuator constants from acquired data is developed, and a CAD model is drafted from measured parameters and used for estimating moments of inertia.
- Design of cascaded-PID control to stabilize the single-tilt tricopter UAV, tested first in simulation and later validated with trial experiments on the developed hardware. This was done to show the feasibility and applicability of the methods.
- Development of hardware for a multi-tilt tricopter vehicle with three independently tilting rotors using open source tools, to yield a fully actuated vehicle. The tilt action of the rotors was made possible by use of a custom-made tilt mechanism for the airframe.

- Development of trifilar experimental identification platform together with an experiment to obtain model parameters of the multi-tilt tricopter from experimental data.
- Design, simulation and analysis of a quaternion feedback (QFB) controller which uses a model with attitude represented in unit quaternions avoiding the gimbal lock problem of Euler angles.
- Design simulation and analysis of linear Model Predictive Controller for the multi-tilt tricopter system using a Linear Parameter Varying (LPV) model of the tricopter which captures the nonlinearities of the tricopter, rather than classic linearisation methods based on operating points which leads to less precision when the system moves further away from the operating point.
- Design of sliding mode controllers to linearize the inner-loop of the nonlinear multi-tilt tricopter UAV. This yields an SMC-linearized closed-loop system with six inputs and outputs corresponding to the multi-tilt tricopter's cartesian positions and attitude which is later characterized as a negative imaginary system, offering a different approach from feedback linearization which is common in literature.
- Identification of NI models for all channels of the SMC-linearized multi-tilt tricopter using the developed closed-loop identification algorithm, and frequency responses of the identified models are used to verify that the models exhibit the NI property.

## 6.2 Directions for Future Research

Although the set aims of this thesis have been achieved, some possible improvements and future research directions for problems tackled in this thesis are outlined below:

- Although optical flow was setup and used for indoor positioning of the single-tilt tricopter as a more affordable alternative to Motion Capture systems (such as MOCAP), the state estimation was not precise and had several errors. It is therefore necessary to develop an alternative indoor navigation system as this

is a key problem especially for researchers who want to perform indoor UAV experiments safely on manageable budgets.

- Even though the system identification method developed to characterize NI system yields satisfactory performance, the precision of the identified model varies with the selection of the Laguerre filter parameter  $p$ . Currently, this parameter is not chosen via a systematic approach but through trial and error making the process a bit tedious. It is therefore necessary to extend this method to provide a more established method for choosing the parameter  $p$ .
- In the NI-based formation control scheme presented, obstacle avoidance and collision have not been considered and these are important considerations for such systems. It is therefore an open problem which offers opportunities for future research.

# Bibliography

- [1] K. P. Valavanis, *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*. Springer Publishing Company, Incorporated, 1st ed., 2007.
- [2] W. Greenfield and D. Greenwood, *Principles of Dynamics*. Prentice-Hall international series in dynamics, Prentice-Hall, 1988.
- [3] B. He, S. Wang, and Y. Liu, “Underactuated robotics: A review,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 4, p. 1729881419862164, 2019.
- [4] B. Crowther, A. Lanzon, M. Maya-Gonzalez, and D. Langkamp, “Kinematic analysis and control design for a non planar multirotor vehicle,” *AIAA Journal of Guidance, Control and Dynamics*, vol. 34, no. 4, pp. 1157–1171, 2011.
- [5] B. Zhao, B. Xian, Y. Zhang, and X. Zhang, “Nonlinear robust adaptive tracking control of a quadrotor uav via immersion and invariance methodology,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 5, pp. 2891–2902, 2015.
- [6] A. Lanzon, A. Freddi, and S. Longhi, “Flight Control of a Quadrotor Vehicle Subsequent to a Rotor Failure,” *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 2, pp. 580–591, 2014.
- [7] A. Alaimo, V. Artale, C. Milazzo, A. Ricciardello, and L. Trefiletti, “Mathematical modeling and control of a hexacopter,” in *2013 International Conference on Unmanned Aircraft Systems, ICUAS 2013 - Conference Proceedings*, pp. 1043–1050, IEEE, 2013.
- [8] D. Brescianini and R. D’Andrea, “Design, modeling and control of an omnidirectional aerial vehicle,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3261–3266, 2016.

- [9] R. Falconi and C. Melchiorri, “Dynamic model and control of an over-actuated quadrotor uav,” *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 192–197, 2012. 10th IFAC Symposium on Robot Control.
- [10] J. Escareño, A. Sanchez, O. Garcia, and R. Lozano, “Triple tilting rotor mini-UAV: Modeling and embedded control of the attitude,” in *Proceedings of the American Control Conference*, (Seattle, USA), pp. 3476–3481, June 2008.
- [11] R. Huang, Y. Liu, and J. J. Zhu, “Guidance, Navigation, and Control System Design for Tripropeller Vertical-Take-Off-and-Landing Unmanned Air Vehicle,” *Journal of Aircraft*, vol. 46, no. 6, pp. 1837–1856, 2009.
- [12] S. Salazar-Cruz, R. Lozano, and J. Escareño, “Stabilization and nonlinear control for a novel trirotor mini-aircraft,” *Control Engineering Practice*, vol. 17, no. 8, pp. 886–894, 2009.
- [13] Duc Anh Ta, I. Fantoni, and R. Lozano, “Modeling and control of a tilt tri-rotor airplane,” in *Proceedings of American Control Conference*, (Montreal, Canada), pp. 131–136, June 2014.
- [14] B. Xian and W. Hao, “Nonlinear robust fault-tolerant control of the tilt trirotor uav under rear servo’s stuck fault: Theory and experiments,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2158–2166, 2019.
- [15] S. Jatsun, O. Emelyanova, A. S. Martinez Leon, and S. Stykanyova, “Control flight of a uav type tricopter with fuzzy logic controller,” in *2017 Dynamics of Systems, Mechanisms and Machines (Dynamics)*, (Omsk, Russia), pp. 1–5, November 2017.
- [16] A. Prach and E. Kayacan, “An MPC-based position controller for a tilt-rotor tricopter VTOL UAV,” *Optimal control applications and methods*, vol. 39, no. 1, pp. 343–356, 2018.
- [17] M. Mehndiratta and E. Kayacan, “Online learning-based receding horizon control of tilt-rotor tricopter: A cascade implementation,” in *Proceedings of the American Control Conference*, (Milwaukee WI, USA), pp. 6378–6383, June 2018.

- [18] H. K. Tran, J. Chiou, N. T. Nam, and V. Tuyen, “Adaptive fuzzy control method for a single tilt tricopter,” *IEEE Access*, vol. 7, pp. 161741–161747, 2019.
- [19] Dronecode-PX4, “PX4 User Guide,” Updated Mar. 2021. [Online]. *Accessed 25th March, 2021 from* <http://docs.px4.io/master/en/>.
- [20] M. Kara Mohamed and A. Lanzon, “Design and control of novel tri-rotor UAV,” in *Proceedings of UKACC International Conference on Control*, (Cardiff, UK), pp. 304–309, September 2012.
- [21] D. Kastelan, M. Konz, and J. Rudolph, “Fully actuated tricopter with pilot-supporting control,” in *Proceedings of the 1st IFAC Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles (ACNAAV)*, vol. 48, (Seville, Spain), pp. 79–84, June 2015.
- [22] E. Servais, B. d’Andréa Novel, and H. Mounier, “Ground control of a hybrid tricopter,” in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 945–950, 2015.
- [23] K.-J. Nam, J. Joung, and D. Har, “Tri-copter uav with individually tilted main wings for flight maneuvers,” *IEEE Access*, vol. 8, pp. 46753–46772, 2020.
- [24] J. Diebel, “Representing attitude: Euler angles, unit quaternions, and rotation vectors,” 2006.
- [25] N. Wei and C. Daizhan, “Leader-following consensus of multi-agent systems under fixed and switching topologies,” *Systems & Control Letters*, vol. 59, no. 3, pp. 209–217, 2010.
- [26] H. A. Poonawala, A. C. Satıcı, H. Eckert, and M. W. Spong, “Collision-free formation control with decentralized connectivity preservation for nonholonomic-wheeled mobile robots,” *IEEE Transactions on Control of Network Systems*, vol. 2, no. 2, pp. 122–130, 2015.
- [27] A. Nagendran, W. Crowther, M. Turner, A. Lanzon, and R. Richardson, “Design, control, and performance of the ‘weed’ 6 wheel robot in the uk mod grand challenge,” *Advanced Robotics*, vol. 28, no. 4, pp. 203–218, 2014.

- [28] G. Foderaro, P. Zhu, H. Wei, T. A. Wettergren, and S. Ferrari, “Distributed optimal control of sensor networks for dynamic target tracking,” *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 142–153, 2018.
- [29] R. Deng, S. He, and J. Chen, “An online algorithm for data collection by multiple sinks in wireless-sensor networks,” *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 93–104, 2018.
- [30] J. Hu, P. Bhowmick, and A. Lanzon, “Distributed adaptive time-varying group formation tracking for multi-agent systems with multiple leaders on directed graphs,” *IEEE Transactions on Control of Network Systems*, vol. 7, no. 1, pp. 140–150, 2020.
- [31] J. Wang, A. Lanzon, and I. R. Petersen, “Robust output feedback consensus for networked negative-imaginary systems,” *IEEE Transactions on Automatic Control*, vol. 60, pp. 2547–2552, Sep. 2015.
- [32] J. Wang, A. Lanzon, and I. R. Petersen, “Robust cooperative control of multiple heterogeneous negative-imaginary systems,” *Automatica*, vol. 61, pp. 64–72, 2015.
- [33] Z. Li, Z. Duan, W. Ren, and G. Feng, “Containment control of linear multi-agent systems with multiple leaders of bounded inputs using distributed continuous controllers,” *International Journal of Robust and Nonlinear Control*, vol. 25, pp. 2101–2121, 05 2015.
- [34] S. Zuo, Y. Song, F. L. Lewis, and A. Davoudi, “Time-varying output formation containment of general linear homogeneous and heterogeneous multiagent systems,” *IEEE Transactions on Control of Network Systems*, vol. 6, no. 2, pp. 537–548, 2019.
- [35] S. Zhao and D. Zelazo, “Translational and scaling formation maneuver control via a bearing-based approach,” *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 429–438, 2017.
- [36] M. H. Trinh, S. Zhao, Z. Sun, D. Zelazo, B. D. O. Anderson, and H.-S. Ahn, “Bearing-based formation control of a group of agents with leader-first follower

- structure,” *IEEE Transactions on Automatic Control*, vol. 64, no. 2, pp. 598–613, 2019.
- [37] C. Wang, H. Thunay, Z. Zuo, B. Lennox, and Z. Ding, “Fixed-time formation control of multirobot systems: Design and experiments,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 8, pp. 6292–6301, 2019.
- [38] I. Bayezit and B. Fidan, “Distributed cohesive motion control of flight vehicle formations,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5763–5772, 2012.
- [39] X. Dong, B. Yu, Z. Shi, and Y. Zhong, “Time-varying formation control for unmanned aerial vehicles: Theories and applications,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 340–348, 2014.
- [40] J. Alonso-Mora, S. Baker, and D. Rus, “Multi-robot formation control and object transport in dynamic environments via constrained optimization,” *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.
- [41] V. P. Tran, F. Santoso, M. A. Garratt, and S. G. Anavatti, “Distributed artificial neural networks-based adaptive strictly negative imaginary formation controllers for unmanned aerial vehicles in time-varying environments,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 3910–3919, 2021.
- [42] J. Hu, B. Lennox, and F. Arvin, “Robust formation control for networked robotic systems using negative imaginary dynamics,” *under review in Automatica*, pp. 1–13, 2021.
- [43] G. Vinnicombe, *Uncertainty and Feedback:  $\mathcal{H}_\infty$  Loop-shaping and the  $\nu$ -gap Metric*. Imperial College Press, 2001.
- [44] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*. Feher/Prentice Hall Digital and, Prentice Hall, 1996.
- [45] H. K. Khalil, *Nonlinear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 2nd ed., 1996.



- [46] A. Lanzon and I. R. Petersen, “A modified positive-real type stability condition,” in *Proceedings of the 2007 European Control Conference*, pp. 3912–3918, 2007.
- [47] A. Lanzon and I. R. Petersen, “Stability robustness of a feedback interconnection of systems with negative imaginary frequency response,” *IEEE Transactions on Automatic Control*, vol. 53, pp. 1042–1046, May 2008.
- [48] M. A. Mabrok, A. G. Kallapur, I. R. Petersen, and A. Lanzon, “Generalizing negative imaginary systems theory to include free body dynamics: Control of highly resonant structures with free body motion,” *IEEE Transactions on Automatic Control*, vol. 59, pp. 2692–2707, Oct 2014.
- [49] B. Bhikkaji, S. O. Reza Moheimani, and I. R. Petersen, “A negative imaginary approach to modeling and control of a collocated structure,” *IEEE/ASME Transactions on Mechatronics*, vol. 17, pp. 717–727, Aug 2012.
- [50] I. A. Mahmood, S. O. R. Moheimani, and B. Bhikkaji, “A new scanning method for fast atomic force microscopy,” *IEEE Transactions on Nanotechnology*, vol. 10, pp. 203–216, March 2011.
- [51] C. Cai and G. Hagen, “Stability analysis for a string of coupled stable subsystems with negative imaginary frequency response,” *IEEE Transactions on Automatic Control*, vol. 55, pp. 1958–1963, Aug 2010.
- [52] J. Wang, A. Lanzon, and I. R. Petersen, “Robust output feedback consensus for multiple heterogeneous negative-imaginary systems,” in *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 2371–2376, Dec 2015.
- [53] J. Wang, A. Lanzon, and I. R. Petersen, “Robust output feedback consensus for networked negative-imaginary systems,” *IEEE Transactions on Automatic Control*, vol. 60, pp. 2547–2552, Sept 2015.
- [54] A. Lanzon and H.-J. Chen, “Feedback stability of negative imaginary systems,” *IEEE Transactions on Automatic Control*, vol. 62, pp. 5620–5633, Nov 2017.
- [55] V. Tran, M. Garratt, and I. Petersen, “Multi-vehicle formation control and obstacle avoidance using negative-imaginary systems theory,” *IFAC Journal of Systems and Control*, vol. 15, p. 100117, 03 2021.

- [56] V. P. Tran, M. Garratt, and I. R. Petersen, “Switching time-invariant formation control of a collaborative multi-agent system using negative imaginary systems theory,” *Control Engineering Practice*, vol. 95, p. 104245, 2020.
- [57] V. P. Tran, F. Santoso, M. A. Garratt, and I. R. Petersen, “Adaptive second-order strictly negative imaginary controllers based on the interval type-2 fuzzy self-tuning systems for a hovering quadrotor with uncertainties,” *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 1, pp. 11–20, 2020.
- [58] O. Skeik, J. Hu, F. Arvin, and A. Lanzon, “Cooperative control of integrator negative imaginary systems with application to rendezvous multiple mobile robots,” pp. 15–20, In: Proceedings of 12th International Workshop on Robot Motion and Control, July 2019.
- [59] B. Haverkamp, C. Chou, M. Verhaegen, and R. Johansson, “Identification of continuous-time mimo state space models from sampled data, in the presence of process and measurement noise,” in *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 2, pp. 1539–1544, 1996.
- [60] R. Mohd-Mokhtar and L. Wang, “Continuous time state space model identification using closed-loop data,” in *2008 Second Asia International Conference on Modelling Simulation (AMS)*, pp. 812–817, 2008.
- [61] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, USA: Cambridge University Press, second ed., 2012.
- [62] C. Meyer, *Matrix Analysis and Applied Linear Algebra*. Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2000.
- [63] B. Stevens, F. Lewis, and E. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. Hoboken, New Jersey: John Wiley and Sons, 2015.
- [64] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. New Jersey, USA: Princeton University Press, 1st ed., 2010.

- [65] W. Ren, R. W. Beard, and E. M. Atkins, “Information consensus in multivehicle cooperative control,” *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.
- [66] J. Hu, P. Bhowmick, F. Arvin, A. Lanzon, and B. Lennox, “Cooperative control of heterogeneous connected vehicle platoons: An adaptive leader-following approach,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 977–984, 2020.
- [67] L. Ji, Y. Shi, and C. Zhang, “Multi-group consensus for heterogeneous agents in cooperative–competitive networks via pinning and adaptive coupling weight methods,” *International Journal of Systems Science*, vol. 53, no. 7, pp. 1469–1482, 2022.
- [68] E. Peymani, H. F. Grip, and A. Saberi, “Homogeneous networks of non-introspective agents under external disturbances -  $H_\infty$  almost synchronization,” *Automatica*, vol. 52, pp. 363–372, 2015.
- [69] M. A. Mabrok, A. G. Kallapur, I. R. Petersen, and A. Lanzon, “Generalizing negative imaginary systems theory to include free body dynamics: Control of highly resonant structures with free body motion,” *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2692–2707, 2014.
- [70] P. Bhowmick and S. Patra, “On decentralized integral controllability of stable negative-imaginary systems and some related extensions,” *Automatica*, vol. 94, pp. 443–451, 2018.
- [71] J. J. Belletrutti and A. G. J. MacFarlane, “Characteristic loci techniques in multivariable-control-system design,” *Proceedings of the Institution of Electrical Engineers*, vol. 118, pp. 1291–1297, Sep 1971.
- [72] A. Macfarlane and J. Belletrutti, “The characteristic locus design method,” *Automatica*, vol. 9, no. 5, pp. 575–588, 1973.
- [73] J. Xiong, I. R. Petersen, and A. Lanzon, “A negative imaginary lemma and the stability of interconnections of linear negative imaginary systems,” *IEEE Transactions on Automatic Control*, vol. 55, pp. 2342–2347, Oct 2010.

- [74] J. B. Kuipers, *Quaternions and rotation sequences: a primer with applications to orbits, aerospace and virtual reality*. Princeton, USA: Princeton University Press, 1999.
- [75] D. Abara, S. Kannan, and A. Lanzon, “Development and stabilization of a single-tilt tricopter with optical flow for GPS-denied environments,” in *Proceedings of the 21st IFAC World Congress*, vol. 53, (Berlin, Germany), pp. 8897–8902, July 2020.
- [76] R. W. Prouty, *Helicopter Performance, Stability and Control*. Malabar, Florida: Krieger Publishing Company, 1995.
- [77] Z. Dydek, A. Annaswamy, and E. Lavretsky, “Adaptive control of quadrotor uavs: A design trade study with flight evaluations,” *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1400–1406, 2013.
- [78] K. Ogata, *Modern Control Engineering*. Instrumentation and controls series, Prentice Hall, 2010.
- [79] D. Abara, P. Bhowmick, and A. Lanzon, “Development and control of multirotor-tilting tricopter with direct actuation for position control,” *under review in International Journal of Control*, pp. 1–31, 2021.
- [80] J. Hu and A. Lanzon, “An innovative tri-rotor drone and associated distributed aerial drone swarm control,” *Robotics and Autonomous Systems*, vol. 103, pp. 162–174, May 2018.
- [81] G. Previati, “Large oscillations of the trifilar pendulum: Analytical and experimental study,” *Mechanism and Machine Theory*, 2021.
- [82] C. M. Harris and A. G. Piersol, *Harris’s Shock and Vibration Handbook*. New York: McGraw-Hill, 2002.
- [83] J. L. du Bois, N. A. J. Lieven, and S. Adhikari, “Error analysis in trifilar inertia measurements,” *Experimental Mechanics*, vol. 49, pp. 533–540, 2009.

- [84] K. Lehmkuhler, K. Wong, and D. Verstraete, “Methods for accurate measurements of small fixed wing uav inertial properties,” *The Aeronautical Journal*, vol. 120, no. 1233, p. 1785–1811, 2016.
- [85] E. Fresk and G. Nikolakopoulos, “Full quaternion based attitude control for a quadrotor,” in *2013 European Control Conference (ECC)*, pp. 3864–3869, 2013.
- [86] B. Wie and P. M. Barba, “Quaternion feedback for spacecraft large angle maneuvers,” *Journal of Guidance, Control and Dynamics*, vol. 8, no. 3, pp. 360–365, 1985.
- [87] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch, “Rigid-body attitude control,” *IEEE Control Systems Magazine*, vol. 31, no. 3, pp. 30–51, 2011.
- [88] H. Kussaba, L. Figueredo, J. Ishihara, and B. Adorno, “Hybrid kinematic control for rigid body pose stabilization using dual quaternions,” *Journal of the Franklin Institute*, vol. 354, pp. 2769–2787, May 2017.
- [89] D. Leith, R. Shorten, W. Leithead, O. Mason, and P. Curran, “Issues in the design of switched linear control systems: A benchmark study,” *International Journal of Adaptive Control and Signal Processing*, vol. 17, no. 2, pp. 103–118, 2013.
- [90] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, “A predictive controller for autonomous vehicle path tracking,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 92–102, 2009.
- [91] B. Biswas, S. Chatterjee, S. Mukherjee, and S. Pal, “A discussion on euler method: a review,” *Electronic Journal of Mathematical Analysis and Applications*, vol. 1, no. 2, pp. 2090–2792, 2013.
- [92] Z. Dostl, *Optimal Quadratic Programming Algorithms: With Applications to Variational Inequalities*. Springer Publishing Company, Incorporated, 1st ed., 2009.
- [93] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems (3rd Ed.)*. USA: Prentice-Hall, Inc., 1997.

- [94] A. Lanzon, A. Freddi, and S. Longhi, “Flight control of a quadrotor vehicle subsequent to a rotor failure,” *AIAA Journal of Guidance, Control and Dynamics*, vol. 37, no. 2, pp. 580–591, 2014.
- [95] Q. Han and X. Liu, “Robust i amp;i adaptive control for a class of quadrotors with disturbances,” *IEEE Access*, vol. 8, pp. 216519–216528, 2020.
- [96] N. E. Gmili, M. Mjahed, A. E. Kari, and H. Ayad, “Particle swarm optimization based proportional-derivative parameters for unmanned tilt-rotor flight control and trajectory tracking,” *Automatika*, vol. 61, no. 2, pp. 189–206, 2020.
- [97] Colmenares-Vázquez, N. Marchand, P. Castillo, J.-E. Gomez-Balderas, J. Álvarez Muñoz, and J. J. Téllez-Guzmán, “Integral backstepping control for trajectory tracking of a hybrid vehicle,” in *Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, (Denver, USA), pp. 209–217, June 2015.
- [98] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, “The Pixhawk Open-Source Computer Vision Framework for Mavs,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, pp. 13–18, September 2011.
- [99] A. I. Hentati, L. Krichen, M. Fourati, and L. C. Fourati, “Simulation Tools, Environments and Frameworks for UAV Systems Performance Analysis,” in *Proceedings of the 14th International Wireless Communications and Mobile Computing Conference*, (Limassol, Cyprus), pp. 1495–1500, June 2018.
- [100] D. Abara, P. Bhowmick, and A. Lanzon, “A negative imaginary robust formation control scheme for a group of networked tricopters over inner-loop sliding-mode control,” *under review in Automatica*, pp. 1–15, 2022.
- [101] R. Mahony, V. Kumar, and P. Corke, “Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor,” *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.
- [102] M. Senanayake, I. Senthooran, J. C. Barca, H. Chung, J. Kamruzzaman, and

- M. Murshed, “Search and tracking algorithms for swarms of robots: A survey,” *Robotics and Autonomous Systems*, vol. 75, pp. 422–434, 2016.
- [103] A. Shukla and H. Karki, “Application of robotics in onshore oil and gas industry—a review part i,” *Robotics and Autonomous Systems*, vol. 75, pp. 490–507, 2016.
- [104] T. P. Nascimento and M. Saska, “Position and attitude control of multi-rotor aerial vehicles: A survey,” *Annual Reviews in Control*, vol. 48, pp. 129–146, 2019.
- [105] M. A. Mabrok, A. G. Kallapur, I. R. Petersen, and A. Lanzon, “Spectral conditions for negative imaginary systems with applications to nanopositioning,” *IEEE/ASME Transactions on Mechatronics*, vol. 19, pp. 895–903, June 2014.
- [106] G. Salcan-Reyes and A. Lanzon, “Negative imaginary synthesis via dynamic output feedback and static state feedback: A riccati approach,” *Automatica*, vol. 104, pp. 220–227, 2019.
- [107] M. Liu, J. Lam, H. Lin, and X. Jing, “Necessary and sufficient conditions on negative imaginarity for interval SISO transfer functions and their interconnection,” *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4362–4368, 2020.
- [108] C. Edwards and S. K. Spurgeon, *Sliding Mode Control: Theory And Applications (1st ed.)*. London: Taylor & Francis, 1998.
- [109] M. A. Mabrok, M. A. Haggag, I. R. Petersen, and A. Lanzon, “A subspace system identification algorithm guaranteeing the negative imaginary property,” in *Proceedings of the 53rd IEEE Conference on Decision and Control*, (Los Angeles, CA, USA), pp. 3180–3185, Dec. 2014.
- [110] L. Ljung, *System Identification: Theory for the User*. Prentice Hall information and system sciences series, Prentice Hall, 1999.
- [111] P. van Overschee and B. de Moor, *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Boston, MA: Springer, 1st ed., 1996.

- [112] J. H. Seo, H. Shim, and J. Back, “Consensus of high-order linear systems using dynamic output feedback compensator: Low gain approach,” *Automatica*, vol. 45, no. 11, pp. 2659–2664, 2009.
- [113] P. J. Antsaklis and A. N. Michel, *A Linear Systems Primer*. Birkhäuser Basel, 1st ed., 2007.
- [114] Dronecode-PX4, “PX4Flow - Smart Camera: Quick Start - Hardware v1.3.” [Online]. Available at [https://github.com/PX4/px4\\_user\\_guide/raw/master/assets/hardware/sensors/px4flow/px4flow\\_manual\\_v1.3.pdf](https://github.com/PX4/px4_user_guide/raw/master/assets/hardware/sensors/px4flow/px4flow_manual_v1.3.pdf).
- [115] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, “An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications,” in *proceedings of the IEEE International Conference on Robotics and Automation*, (Karlsruhe, Germany), pp. pp. 1736–1741, May, May 2013.
- [116] GARMIN, “Lidar-Lite v3 Operation Manual and Technical Specifications,” 2016. [Online]. Available at [http://static.garmin.com/pumac/LIDAR\\_Lite\\_v3\\_Operation\\_Manual\\_and\\_Technical\\_Specifications.pdf](http://static.garmin.com/pumac/LIDAR_Lite_v3_Operation_Manual_and_Technical_Specifications.pdf).
- [117] Matek-Systems, “PDB-XT60 with BEC 5V and 12V.” [Online]. Available at [http://www.mateksys.com/downloads/PDB-XT60\\_Manual\\_EN.pdf](http://www.mateksys.com/downloads/PDB-XT60_Manual_EN.pdf).
- [118] K. W. Eure, C. Quach, S. L. Vazquez, E. F. Hogge, and B. L. Hill, “An Application of UAV Attitude Estimation Using a Low-Cost Inertial Navigation System,” 2013.
- [119] C. Ramirez-Atencia and D. Camacho, “Extending QGroundControl for automated mission planning of Uavs,” *Sensors (Switzerland)*, vol. 18, no. 7, 2018.



# Appendix A

## Platform Description

This appendix provides a generic overview of the hardware and software tools used in the development of the tricopter UAV platforms (single-tilt/classic tricopter and multi-tilt tricopter) which are considered in this thesis. The material here is therefore not specific, and applies to both tricopter platforms. Hence, the reader should note that the term ‘tricopter’ as used within this appendix applies to both the single-tilt/classic and multi-tilt tricopters.

### A.1 Hardware Design

The tricopter consists of a carbon fibre frame to which several hardware components are attached. It makes use of an onboard computer which is the central point of the UAV, responsible for stabilizing the system and is usually called a *flight controller*. For this research, the *Pixhawk* autopilot is used as the flight controller. A high-level wiring diagram of the entire setup for the tricopter is given in Figure A.1. In

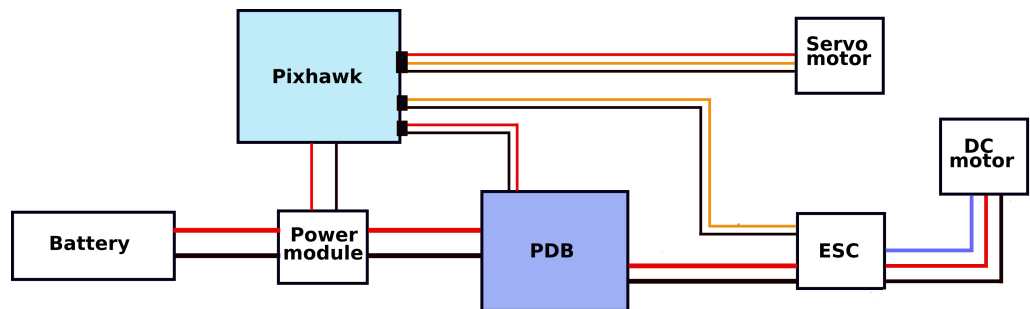


Figure A.1: Block diagram of the setup.

general, Brushless Direct Current (BLDC) motors are used together with propellers for

propulsion. A Lithium Polymer (LiPo) battery is used as the power source, and this is connected via a power module to rectify the supplied battery power to 5 volts for the Pixhawk, while also supplying the battery's voltage to a power distribution board which distributes power to all the Electronic Speed Controllers (ESCs) and hence the motors. The servos are powered directly from the servo output rail of the Pixhawk. The Pixhawk is designed to support power redundancy so it is powered from its main supply using the power module and also via the servo rail from a Power Distribution Board (PDB) as depicted in Figure A.1. The ESCs are used to internally regulate the speed of the motors as is typically done in multicopter systems. The full list of components and their technical specification is given in Appendices B.1 and B.2, for the single-tilt and multi-tilt tricopters respectively.

### A.1.1 Pixhawk Flight Controller

The Pixhawk autopilot was designed by the open-hardware project [98] carried out by a team from the Computer Vision and Geometry lab of ETH Zurich. Several versions of the Pixhawk have been developed over the years, but the first generation of the autopilot (Pixhawk 1) is the one used in this work. It runs a 168MHz 32bit STM32F427 Cortex-M4F processor with 256KB RAM and 2MB Flash. It has a total of 14 PWM/servo outputs, and several ports for connecting additional peripherals including UART, I2C, CAN etc). For sensing, the Pixhawk includes on-board gyroscopes, accelerometers, magnetometers and barometers, with support for a wide range of peripherals. The reader may refer to [19] for a full hardware specification of the Pixhawk.



Figure A.2: Pixhawk 1 Flight controller.

The Pixhawk 1 was chosen primarily because it supports all the necessary features for guidance, navigation and control of several systems including UAVs and yet is relatively easy to setup and customize. Also, on commencement of this project, only

the Pixhawk 1 had some support from Mathworks through a Pilot Support Package (PSP). At the time of writing this report, Mathworks has official support for other Pixhawk-based autopilots such as Pixhawk Cube, Pixhawk 4 and Pixracer. It should be noted that the Pixhawk project has a number of open designs/schematics and all boards based on a particular design should run the same firmware. Each design is named using the designation: FMUvX. For example, FMUv1, FMUv2 etc., where FMU stands for Flight Manangement Unit. Higher FMU numbers indicate a more recent board but not necessarily increased capabilities or features. The choice of autopilot will however depend on the specific requirements of the project to be carried out and the physical constraints/form factor rather than FMU version. The main differences as noted in [19] are seen in the speed and sizes of the memories on each FMU version. For example, Pixhawk Cube uses FMUv3 with 2MB Flash memory while the Pixhawk 1 uses FMUv2 which supports 1MB Flash memory but all other features are identical. Furthermore, the Pixracer supports FMUv4 with increased RAM, faster CPU and more serial ports compared to FMU versions 2 and 3. Besides these, most of its other features are identical to those of FMU versions 2 and 3. Another important point to note is that manufacturers have recently updated the Pixhawk 1 hardware to support 2MB Flash and as a result FMUv2 can be used on these newer Pixhawk1 boards such as the one used in this thesis.

### A.1.2 Additional Sensors

To obtain estimates of the  $x$  and  $y$  states when indoors, an optical flow sensor is used. The PX4Flow camera is an optical flow sensor with a resolution of  $752 \times 480$  pixels and calculates the optical flow at 250Hz [114, 115].

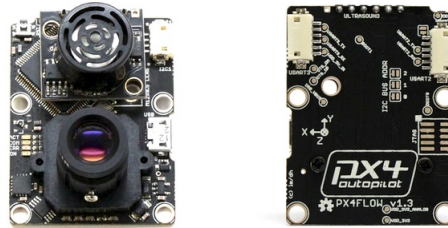


Figure A.3: PX4Flow optical flow sensor.

A top and bottom pictorial view is shown in Figure A.3. It is an ARM Cortex M4 based sensor system which processes optical flow at 250 frames per second and

is designed to work indoors as well as outdoors in low light conditions, making use of a mounted lens and ultrasonic distance sensor. Although the ultrasonic distance sensor on-board the PX4Flow is mostly sufficient, it is recommended by the PX4 community to use an external range-finder, such that the dedicated range-finder computes the height independently allowing all the resources of the PX4Flow to be dedicated to obtaining optical flow measurements only, thereby yielding improved precision in the state estimates. For this purpose, the Lidar-Lite v3 range-finder [116] shown in Figure A.4 was used.



Figure A.4: Lidar-Lite V3 range-finder.

It is a high performance optical distance measurement sensor for unmanned vehicle applications with a range of 40 m (131 ft) weighing about 0.022 Kg only. The Lidar-Lite can be connected to the autopilot using either I2C or PWM which makes it flexible. For more details and full list of technical specification, the reader should see [116].

### A.1.3 Propulsion and Power System

The propulsion system is a standard motor-propeller setup typically used in multicopter systems. A DC motor with a fixed-pitch propeller is mounted on the tricopter frame and powered by an ESC, which regulates the angular speed of the motors based on setpoints from the flight controller outputs. For clean wiring and ease of maintenance, the Matek XT60 PDB was utilized. The XT60 PDB can distribute power from a LiPo pack to up to 6 ESCs, while providing regulated 5V and 12V DC for connecting receivers, cameras and other components (see [117] for full specifications). The PDB utilizes an inbuilt XT60 connector for ease of connecting to a LiPo battery pack. The LiPo battery specifications for each tricopter are given in Appendices B.1 and B.2, for the single-tilt/classic and multi-tilt tricopter UAVs respectively.



Figure A.5: Main Power supply components

#### A.1.4 Heading control concept

Unlike a traditional multicopter which has rotors in pairs (even number of rotors) such that inherent aerodynamic reaction torques are cancelled by using counter-rotating propellers, the tricopter has an odd number of rotors. Consequently, there is an imbalance due to the reaction (drag) torques from the spinning rotors. Hence, to control the heading (yaw angle), a servo is attached to the rotors (only the tail rotor for the single-tilt case) and used to create a resultant balancing torque which compensates for the reaction torques from the spinning rotors. This concept is depicted in Figure A.6.

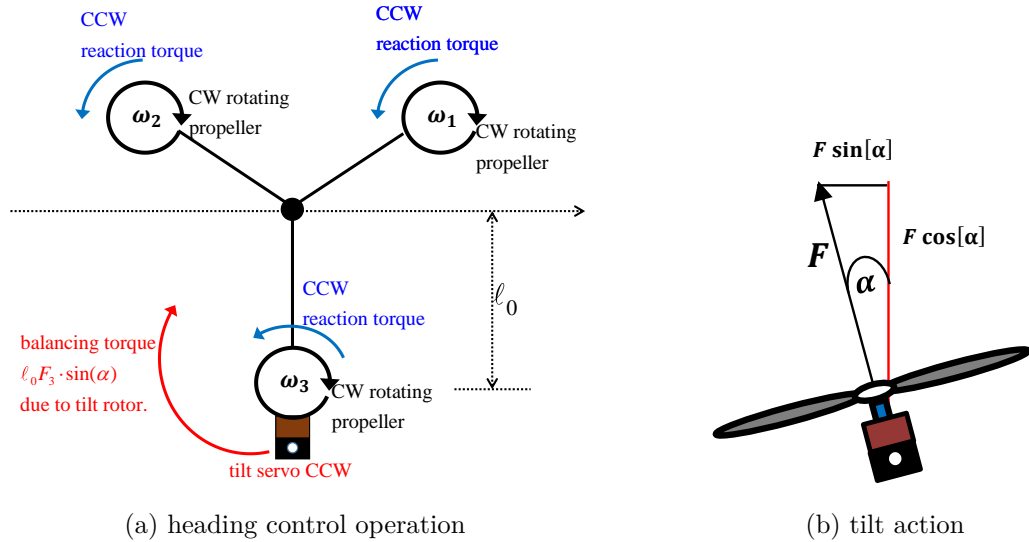


Figure A.6: Heading control; CW and CCW mean Clockwise and Counter clockwise respectively.

As observed from Figure A.6a, all rotors spin clockwise so that counter-clockwise reaction torques are produced. If left at this state, the tricopter will rotate uncontrollably counter-clockwise around the vertical axis (yaw motion or heading will be

unstable). To balance the heading and achieve stable hover, the tail servo will tilt counter-clockwise as depicted in Figure A.6b so that rotor 3 is tilted counter-clockwise thereby producing a balancing torque to compensate for the reaction torques from each rotor.

The final assembly for both tricopters is given in Figures A.7 and A.8 below.

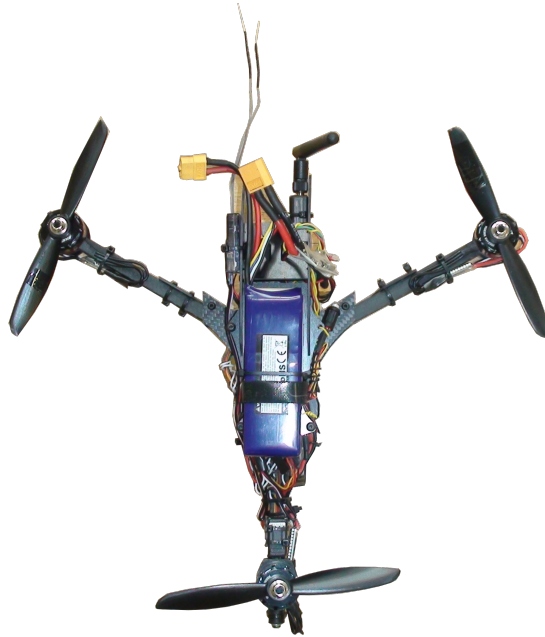


Figure A.7: Single-tilt tricopter Assembly



Figure A.8: Multi-tilt tricopter Assembly

## A.2 Software Description

The Pixhawk controller supports two commonly used flight control software; PX4 and Ardupilot Mega (APM) [118] flight control software. In this work, the PX4 flight control software is used as it was developed mainly for Pixhawk products.

### A.2.1 PX4 Architecture

The current version of the PX4 flight control software consists of two main layers: the *flight stack* and the *middleware* [115]. The flight stack is a collection of guidance, navigation and control algorithms for autonomous drones, with controllers for fixed wing, multirotor and Vertical Take-off and Landing (VTOL) airframes. It also contains estimators for attitude and position. The middleware is a general robotics layer that supports any type of autonomous robot, providing internal and external communications and hardware integration. The PX4 flight control software was chosen because it can be customized for multiple airframes and control allocations.

### A.2.2 QGroundControl Interface

The PX4 vehicle setup and flight control tuning was done using *QGroundControl* (QGC) station software [99, 119] (also see <http://docs.qgroundcontrol.com/en>). QGC contains parameters which control all the settings that can be customized for any autonomous robot application, such as selection of PID gains, ESC and battery calibration, autonomous path planning and sensor calibrations. A screenshot of QGC showing the interface and some customizable parameters is given in Figure A.9.

	GROUPS	Multicopter Attitude Control Parameters			Tools
Summary	Camera trigger	MC_ACRO_P_MAX	360.000 deg/s	Max acro pitch rate	
Firmware	Circuit Breaker	MC_ACRO_R_MAX	360.000 deg/s	Max acro roll rate	
Airframe	Commander	MC_ACRO_Y_MAX	360.000 deg/s	Max acro yaw rate	
Radio	Data Link Loss	MC_PITCHRATE_D	0.003	Pitch rate D gain	
Flight Modes	GPS Failure Navigation	MC_PITCHRATE_FF	0.000	Pitch rate feedforward	
Sensors	Geofence	MC_PITCHRATE_I	0.050	Pitch rate I gain	
Power	Land Detector	MC_PITCHRATE_MAX	220.000 deg/s	Max pitch rate	
Safety	MAVLink	MC_PITCHRATE_P	0.150	Pitch rate P gain	
	Miscellaneous	MC_PITCH_P	7.000 1/s	Pitch P gain	
	Mission	MC_RATT_TH	1.000	Threshold for Rattitude mode	
Parameters	Multicopter Attitude Control	MC_ROLLRATE_D	0.003	Roll rate D gain	
	Multicopter Position Control	MC_ROLLRATE_FF	0.000	Roll rate feedforward	
	PWM Outputs	MC_ROLLRATE_I	0.050	Roll rate I gain	
	Position Estimator INAV	MC_ROLLRATE_MAX	220.000 deg/s	Max roll rate	
	Radio Calibration	MC_ROLLRATE_P	0.150	Roll rate P gain	

Figure A.9: QGroundControl Ground station application.

### A.3 Summary

This brief chapter has provided an overview of the hardware and software for the tricopters considered in this thesis. Firstly, details of the hardware design have been provided, together with relevant information on the key hardware components and concepts used to realise the platforms. Then, the software used for the flight controller and ground station has also been discussed.



# Appendix B

## Tricopter System Specification

### B.1 Single-tilt tricopter

#### B.1.1 Identification

Feature	Description
Drone name	Tri-ST-Exp
UOM ID	UAS-0040-FSE-SOE-EEE
Primary purpose	To test and validate first principles model and custom control allocation with control design. Also used to test flights in GPS-denied environments using optical flow.

#### B.1.2 Performance Summary

	Feature	Description
<b>Speeds</b>	Minimum speed	0 m/s
	Cruise speed	3 m/s (limit set in flight controller)
	Top speed	10 m/s (limit set in flight controller)
<b>Weights</b>	Empty weight <sup>1</sup>	0.56 kg (exc. battery / GPS)
	Take-off weight	0.74 kg (inc. battery)
<b>Limits</b>	Maximum Altitude <sup>2</sup>	40 m (limit set in flight controller)
	Return to Land Altitude	30 m (limit set in flight controller)
	Flying Area radius	50 m
	Projected endurance	10 – 15 minutes

### B.1.3 System Hardware Description

#### B.1.3.1 Airframe

Feature	Description
Original manufacturer <sup>3</sup>	RC-Explorer
Model/Configuration	Tricopter-Y geometry, upward facing
Material	Carbon fibre

#### B.1.3.2 Propulsion System

	Feature	Description
<b>Motors</b>	Motor configuration	three motors upward facing, with the tail one mounted on a servo.
	No. of motors	3
	Manufacturer / Type	Emax / 2207-eco Brushless DC
	Servo configuration	One servo for tail motor
<b>Servos</b>	Manufacturer / Type	KST / DS215MG digital
	No. of propellers	3(3 × CW)
<b>Propeller</b>	Material	Carbon fibre
	Size	6 × 4.5 inches
<b>Battery</b>	Chemistry	Lithium-Polymer (LiPo)
	Cell rating	3S (11.1 Volts)
	Capacity	1800 mAh
	No. of batteries <sup>4</sup>	1
<b>Speed Con- troller</b>	Supplier / Type	Aikon / AK32
	Input voltage	3-6S LiPo.
	Continuous current <sup>5</sup>	35 Amps

<sup>1</sup>**The Empty weight** is the weight of the drone including motors, speed controllers, propeller, telemetry radio, receiver and associated connections. It is also the same as the **Indoor operating weight** which is when the drone is operated on a Gimbal tethered to an external power supply or battery and without the GPS module mounted.

<sup>2</sup>**Maximum Altitude** here is also the maximum height set in the flight controller for geofencing. If for any reason, the drone breaches this height, RETURN mode is activated.

<sup>3</sup>The original airframe has undergone some customizations to achieve a novel airframe concept

<sup>4</sup>This is the number of batteries used for flying.

<sup>5</sup>The maximum amount of continuous current which the ESC can safely handle.

**B.1.3.3 Control System**

	Feature	Description
<b>Radio Con- troller</b>	Manufacturer	FrSky (Transmitter and Receiver)
	Model	Taranis X9D-Plus Transmitter and X8R receiver (Mode D16)
	No. of channels	Up to 16
	Operating frequency	2.4 GHz
<b>Telemetry</b>	Manufacturer	3DR
	Firmware	SiK v1.9
	Operating frequency	433 MHz

**B.1.3.4 Avionics System**

	Feature	Description
<b>Flight Controller</b>	Manufacturer	UnmannedTech
	Model	Pixhawk 1 (with 2 MB flash)
	Firmware <sup>6</sup>	PX4 v1.10.1
	Operating modes	Manual, Stabilize, Altitude Hold, Position
<b>GPS/Compass</b>	Manufacturer	UnmannedTech
	Model	Ublox NEO-M8N
	Firmware	Ublox
<b>Optical flow</b>	Manufacturer	3D-Robotics
	Model	PX4Flow v1.3
	Firmware / Estimator	Px4 / EKF
<b>Range finder</b>	Manufacturer	Garmin
	Model	Lidar-Lite v3
	Range	40m (131ft.)

<sup>6</sup>A stable release of the original PX4 firmware is customized.

## B.2 Multi-tilt tricopter

### B.2.1 Identification

Feature	Description
Drone name	Tri-MT-Talon
UOM ID	UAS-0039-FSE-SOE-EEE
Primary purpose	To validate novel airframe concept with advanced control

### B.2.2 Performance Summary

	Feature	Description
<b>Speeds</b>	Minimum speed	0 m/s
	Cruise speed	3 m/s (limit set in flight controller)
	Top speed	10 m/s (limit set in flight controller)
<b>Weights</b>	Empty weight	1.0 kg (exc. battery / GPS)
	Take-off weight	1.23 kg (inc. battery)
<b>Limits</b>	Maximum Altitude	40 m (limit set in flight controller)
	Return to Land Altitude	30 m (limit set in flight controller)
	Flying Area radius	50 m
	Projected endurance	10 – 15 minutes

### B.2.3 System Hardware Description

#### B.2.3.1 Airframe

Feature	Description
Original manufacturer <sup>7</sup>	Turnigy
Model/Configuration	Tricopter-Y geometry, upward facing
Material	Carbon fibre

<sup>7</sup>The original airframe has undergone some customizations to achieve a novel airframe concept

**B.2.3.2 Propulsion System**

	Feature	Description
<b>Motors</b>	Motor configuration	upward facing, mounted on servos in range $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$
	No. of motors / servos	3
	Manufacturer / Type	Turnigy / D2836-11 Brushless Outrunner
	Power per motor	220 Watts
<b>Propeller</b>	No. of propellers	$3(3 \times \text{CW})$
	Material	Carbon fibre
	Size	$10 \times 4.5$ inches
<b>Battery</b>	Chemistry	Lithium-Polymer (LiPo)
	Cell rating	3S (11.1 Volts)
	Capacity	1800 mAh
	No. of batteries	1
<b>Speed Controller</b>	Supplier	Chaos
	Input voltage	3-6S LiPo.
	Continuous current	30 Amps

**B.2.3.3 Control System**

	Feature	Description
<b>Radio Controller</b>	Manufacturer	FrSky (Transmitter and Receiver)
	Model	Taranis X9D-Plus Transmitter and X8R receiver (Mode D16)
	No. of channels	Up to 16
	Operating frequency	2.4 GHz
<b>Telemetry</b>	Manufacturer	3DR
	Firmware	SiK v1.9
	Operating frequency	433 MHz

**B.2.3.4 Avionics System**

	<b>Feature</b>	<b>Description</b>
<b>Flight Controller</b>	Manufacturer	UnmannedTech
	Model	Pixhawk 1 (with 2 MB flash)
	Firmware <sup>8</sup>	PX4 v1.10.1
	Operating modes	Manual, Stabilize, Altitude Hold, Position
<b>GPS/Compass</b>	Manufacturer	UnmannedTech
	Model	Ublox NEO-M8N
	Firmware	Ublox

---

<sup>8</sup>A stable release of the original PX4 firmware is customized.

# Appendix C

## MPC Formulation

### C.1 Forward Euler method of discretization

Given an initial time condition  $t_k$  and initial state  $x_k$ , the forward Euler method is given by the difference equation [91],

$$y_{k+1} = y_k + \dot{y}_k \Delta t.$$

where  $\Delta t$  is the sample time. Thus,

$$\dot{y}_k = \frac{y_{k+1} - y_k}{\Delta t}. \quad (\text{C.1})$$

Consider the following linear time invariant system,

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y} &= \mathbf{C}\mathbf{x}(t). \end{aligned} \quad (\text{C.2})$$

The derivative  $\dot{\mathbf{x}}$  may be represented using (C.1) as

$$\begin{aligned} \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\Delta t} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \\ \dot{\mathbf{x}}_{k+1} &= \dot{\mathbf{x}}_k + \mathbf{A}\mathbf{x}_k \Delta t + \mathbf{B}\mathbf{u}_k \Delta t, \\ \dot{\mathbf{x}}_{k+1} &= (\mathbf{I} + \mathbf{A}\Delta t)\mathbf{x}_k + \mathbf{B}\Delta t\mathbf{u}_k, \\ \dot{\mathbf{x}}_{k+1} &= \mathbf{A}^d\mathbf{x}_k + \mathbf{B}^d \end{aligned}$$

where  $\mathbf{A}^d = \mathbf{I} + \mathbf{A}\Delta t$ ,  $\mathbf{B}^d = \mathbf{B}\Delta t$  and the matrix  $\mathbf{C}$  of (C.2) is unchanged. By applying these results to (4.15),

$$\mathbf{A}^d(\dot{\phi}, \dot{\theta}) = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \frac{J_y - J_z}{J_x} \dot{\theta} \Delta t \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \frac{J_z - J_x}{J_y} \dot{\phi} \Delta t \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & \frac{J_x - J_y}{2J_z} \dot{\theta} \Delta t & 0 & \frac{J_x - J_y}{2J_z} \dot{\phi} \Delta t & 0 & 1 \end{bmatrix}, \text{ and } \mathbf{B}^d = \begin{bmatrix} 0 & 0 & 0 \\ \frac{\Delta t}{J_x} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{\Delta t}{J_y} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{\Delta t}{J_z} \end{bmatrix}.$$

## C.2 Cost function simplification

### C.2.1 Part I

Given the cost function as

$$\begin{aligned} J &= \min_{\Delta \mathbf{u}_k} \frac{1}{2} \sum_{i=0}^{N-1} \left( \mathbf{e}_{k+i}^\top \mathbf{V} \mathbf{e}_{k+i} + \Delta \mathbf{u}_{k+i}^\top \mathbf{W} \Delta \mathbf{u}_{k+i} \right) + \frac{1}{2} \mathbf{e}_{k+N}^\top \mathbf{Z} \mathbf{e}_{k+N}, \quad (4.16 \text{ revisited}) \\ &= \min_{\Delta \mathbf{u}_k} \sum_{i=0}^{N-1} (\text{LHS}) + \text{RHS} \end{aligned}$$

Let  $\mathbf{e}_k = \mathbf{r}_k - \tilde{\mathbf{C}} \tilde{\mathbf{x}}_k$ , then expanding the LHS yields:

$$\begin{aligned} \text{LHS} &= \frac{1}{2} \left( \mathbf{e}_{k+i}^\top \mathbf{V} \mathbf{e}_{k+i} + \Delta \mathbf{u}_{k+i}^\top \mathbf{W} \Delta \mathbf{u}_{k+i} \right), \\ &= \frac{1}{2} \left[ \left( \mathbf{r}_{k+i}^\top - \tilde{\mathbf{x}}_{k+1}^\top \tilde{\mathbf{C}}^\top \right) \mathbf{V} \left( \mathbf{r}_{k+i} - \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+1} \right) + \Delta \mathbf{u}_{k+i}^\top \mathbf{W} \Delta \mathbf{u}_{k+i} \right], \\ &= \frac{1}{2} \mathbf{r}_{k+i}^\top \mathbf{V} \mathbf{r}_{k+i} - \frac{1}{2} \mathbf{r}_{k+i}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+1} - \frac{1}{2} \tilde{\mathbf{x}}_{k+1}^\top \tilde{\mathbf{C}}^\top \mathbf{V} \mathbf{r}_{k+i} + \frac{1}{2} \tilde{\mathbf{x}}_{k+1}^\top \tilde{\mathbf{C}}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+1} \\ &\quad + \frac{1}{2} \Delta \mathbf{u}_{k+i}^\top \mathbf{W} \Delta \mathbf{u}_{k+i}, \\ &= \frac{1}{2} \mathbf{r}_{k+i}^\top \mathbf{V} \mathbf{r}_{k+i} - \frac{1}{2} \mathbf{r}_{k+i}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+1} - \frac{1}{2} \left( \mathbf{r}_{k+i}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+1} \right)^\top + \frac{1}{2} \tilde{\mathbf{x}}_{k+1}^\top \tilde{\mathbf{C}}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+1} \\ &\quad + \frac{1}{2} \Delta \mathbf{u}_{k+i}^\top \mathbf{W} \Delta \mathbf{u}_{k+i}, \end{aligned}$$

The term  $\mathbf{r}_{k+i}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+1} \in \mathbb{R}^{1 \times 1}$  so that,

$$\text{LHS} = \frac{1}{2} \mathbf{r}_{k+i}^\top \mathbf{V} \mathbf{r}_{k+i} - \mathbf{r}_{k+i}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+1} + \frac{1}{2} \tilde{\mathbf{x}}_{k+1}^\top \tilde{\mathbf{C}}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+1} + \frac{1}{2} \Delta \mathbf{u}_{k+i}^\top \mathbf{W} \Delta \mathbf{u}_{k+i}. \quad (\text{C.3})$$

Similarly,

$$\text{RHS} = \frac{1}{2} \mathbf{e}_{k+N}^\top \mathbf{Z} \mathbf{e}_{k+N},$$



$$\begin{aligned}
&= \frac{1}{2} \left[ \left( \mathbf{r}_{k+N} - \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N} \right)^\top \mathbf{Z} \left( \mathbf{r}_{k+N} - \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N} \right) \right], \\
&= \frac{1}{2} \left[ \left( \mathbf{r}_{k+N}^\top - \tilde{\mathbf{x}}_{k+N}^\top \tilde{\mathbf{C}}^\top \right) \mathbf{Z} \left( \mathbf{r}_{k+N} - \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N} \right) \right], \\
&= \frac{1}{2} \mathbf{r}_{k+N}^\top \mathbf{Z} \mathbf{r}_{k+N} - \frac{1}{2} \mathbf{r}_{k+N}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N} - \frac{1}{2} \tilde{\mathbf{x}}_{k+N}^\top \tilde{\mathbf{C}}^\top \mathbf{Z} \mathbf{r}_{k+N} + \frac{1}{2} \tilde{\mathbf{x}}_{k+N}^\top \tilde{\mathbf{C}}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N}, \\
&= \frac{1}{2} \mathbf{r}_{k+N}^\top \mathbf{Z} \mathbf{r}_{k+N} - \frac{1}{2} \mathbf{r}_{k+N}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N} - \frac{1}{2} \left( \mathbf{r}_{k+N}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N} \right)^\top + \frac{1}{2} \tilde{\mathbf{x}}_{k+N}^\top \tilde{\mathbf{C}}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N}, \\
&= \frac{1}{2} \mathbf{r}_{k+N}^\top \mathbf{Z} \mathbf{r}_{k+N} - \mathbf{r}_{k+N}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N} + \frac{1}{2} \tilde{\mathbf{x}}_{k+N}^\top \tilde{\mathbf{C}}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N}, \tag{C.4}
\end{aligned}$$

since the term  $\mathbf{r}_{k+N}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N} \in \mathbb{R}^{1 \times 1}$  which is a scalar quantity. Combining the LHS (C.3) and RHS (C.4) the cost function (4.16) becomes,

$$\begin{aligned}
J &= \min_{\Delta \mathbf{u}_k} \sum_{i=0}^{N-1} \left[ \frac{1}{2} \mathbf{r}_{k+i}^\top \mathbf{V} \mathbf{r}_{k+i} - \mathbf{r}_{k+i}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+i} + \frac{1}{2} \tilde{\mathbf{x}}_{k+i}^\top \tilde{\mathbf{C}}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+i} + \frac{1}{2} \Delta \mathbf{u}_{k+i}^\top \mathbf{W} \Delta \mathbf{u}_{k+i} \right] \\
&\quad + \frac{1}{2} \mathbf{r}_{k+N}^\top \mathbf{Z} \mathbf{r}_{k+N} - \frac{1}{2} \mathbf{r}_{k+N}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N} + \frac{1}{2} \tilde{\mathbf{x}}_{k+N}^\top \tilde{\mathbf{C}}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N}, \\
&= \min_{\Delta \mathbf{u}_k} \sum_{i=1}^{N-1} \left[ -\mathbf{r}_{k+i}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+i} + \frac{1}{2} \tilde{\mathbf{x}}_{k+i}^\top \tilde{\mathbf{C}}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+i} \right] + \sum_{i=0}^{N-1} \left[ \frac{1}{2} \Delta \mathbf{u}_{k+i}^\top \mathbf{W} \Delta \mathbf{u}_{k+i} \right] \\
&\quad - \frac{1}{2} \mathbf{r}_{k+N}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N} + \frac{1}{2} \tilde{\mathbf{x}}_{k+N}^\top \tilde{\mathbf{C}}^\top \mathbf{Z} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_{k+N} + c_1. \tag{4.20 revisited}
\end{aligned}$$

where

$$c_1 = \frac{1}{2} \left( \mathbf{r}_{k+1}^\top \mathbf{V} \mathbf{r}_{k+1} + \mathbf{r}_{k+N}^\top \mathbf{Z} \mathbf{r}_{k+N} \right) - \mathbf{r}_k^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_k + \frac{1}{2} \tilde{\mathbf{x}}_k^\top \tilde{\mathbf{C}}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_k. \tag{C.5}$$

The terms  $-\mathbf{r}_k^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_k$  and  $\frac{1}{2} \tilde{\mathbf{x}}_k^\top \tilde{\mathbf{C}}^\top \mathbf{V} \tilde{\mathbf{C}} \tilde{\mathbf{x}}_k$  in (C.5) are from the LHS when  $i = 0$  implying the current/present state which is known and not predicted, and is hence, a constant offset.

### C.2.2 Part II - Future State Prediction Formula

Given the augmented system,

$$\begin{aligned}
\tilde{\mathbf{x}}_{k+1} &= \tilde{\mathbf{A}}_{k+1} \tilde{\mathbf{x}}_k + \tilde{\mathbf{B}} \Delta \mathbf{u}_k, \\
\tilde{\mathbf{y}}_k &= \tilde{\mathbf{C}} \tilde{\mathbf{x}}_k,
\end{aligned} \tag{4.19 revisited}$$

with prediction horizon  $N = 4$ . If the current state is  $\tilde{\mathbf{x}}_{k=0}$ , then the MPC predicts 4 states in the future from the current state as follows:

$$\begin{aligned}
\tilde{\mathbf{x}}_1 &= \tilde{\mathbf{A}}_1 \tilde{\mathbf{x}}_0 + \tilde{\mathbf{B}} \Delta \mathbf{u}_0, \\
\tilde{\mathbf{x}}_2 &= \tilde{\mathbf{A}}_2 \tilde{\mathbf{x}}_1 + \tilde{\mathbf{B}} \Delta \mathbf{u}_1, \\
&= \tilde{\mathbf{A}}_2 [\tilde{\mathbf{A}}_1 \tilde{\mathbf{x}}_0 + \tilde{\mathbf{B}} \Delta \mathbf{u}_0] + \tilde{\mathbf{B}} \Delta \mathbf{u}_1, \\
&= \tilde{\mathbf{A}}_2 \tilde{\mathbf{A}}_1 \tilde{\mathbf{x}}_0 + \tilde{\mathbf{A}}_2 \tilde{\mathbf{B}} \Delta \mathbf{u}_0 + \tilde{\mathbf{B}} \Delta \mathbf{u}_1. \\
&\vdots \\
\tilde{\mathbf{x}}_4 &= \tilde{\mathbf{A}}_4 \tilde{\mathbf{x}}_3 + \tilde{\mathbf{B}} \Delta \mathbf{u}_3, \\
&= \tilde{\mathbf{A}}_4 [\tilde{\mathbf{A}}_3 \tilde{\mathbf{A}}_2 \tilde{\mathbf{A}}_1 \tilde{\mathbf{x}}_0 + \tilde{\mathbf{A}}_3 \tilde{\mathbf{A}}_2 \tilde{\mathbf{B}} \Delta \mathbf{u}_0 + \tilde{\mathbf{A}}_3 \tilde{\mathbf{B}} \Delta \mathbf{u}_1 + \tilde{\mathbf{B}} \Delta \mathbf{u}_2] + \tilde{\mathbf{B}} \Delta \mathbf{u}_3, \\
&= \tilde{\mathbf{A}}_4 \tilde{\mathbf{A}}_3 \tilde{\mathbf{A}}_2 \tilde{\mathbf{A}}_1 \tilde{\mathbf{x}}_0 + \tilde{\mathbf{A}}_4 \tilde{\mathbf{A}}_3 \tilde{\mathbf{A}}_2 \tilde{\mathbf{B}} \Delta \mathbf{u}_0 + \tilde{\mathbf{A}}_4 \tilde{\mathbf{A}}_3 \tilde{\mathbf{B}} \Delta \mathbf{u}_1 + \tilde{\mathbf{A}}_4 \tilde{\mathbf{B}} \Delta \mathbf{u}_2 + \tilde{\mathbf{B}} \Delta \mathbf{u}_3.
\end{aligned}$$

More compactly,

$$\begin{aligned}
\begin{bmatrix} \tilde{\mathbf{x}}_1 \\ \tilde{\mathbf{x}}_2 \\ \tilde{\mathbf{x}}_3 \\ \tilde{\mathbf{x}}_4 \end{bmatrix} &= \begin{bmatrix} \tilde{\mathbf{A}}_1 \\ \tilde{\mathbf{A}}_2 \tilde{\mathbf{A}}_1 \\ \tilde{\mathbf{A}}_3 \tilde{\mathbf{A}}_2 \tilde{\mathbf{A}}_1 \\ \tilde{\mathbf{A}}_4 \tilde{\mathbf{A}}_3 \tilde{\mathbf{A}}_2 \tilde{\mathbf{A}}_1 \end{bmatrix} \tilde{\mathbf{x}}_{k=0} + \begin{bmatrix} \tilde{\mathbf{B}} & \dots & \dots & \mathbf{0} \\ \tilde{\mathbf{A}}_2 \tilde{\mathbf{B}} & \tilde{\mathbf{B}} & & \vdots \\ \tilde{\mathbf{A}}_3 \tilde{\mathbf{A}}_2 \tilde{\mathbf{B}} & \tilde{\mathbf{A}}_3 \tilde{\mathbf{B}} & \tilde{\mathbf{B}} & \\ \tilde{\mathbf{A}}_4 \tilde{\mathbf{A}}_3 \tilde{\mathbf{A}}_2 \tilde{\mathbf{B}} & \tilde{\mathbf{A}}_4 \tilde{\mathbf{A}}_3 \tilde{\mathbf{B}} & \tilde{\mathbf{A}}_4 \tilde{\mathbf{B}} & \tilde{\mathbf{B}} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_0 \\ \Delta \mathbf{u}_1 \\ \Delta \mathbf{u}_2 \\ \Delta \mathbf{u}_3 \end{bmatrix}, \\
&\Leftrightarrow \tilde{\mathbf{x}} = \bar{\mathbf{A}} \tilde{\mathbf{x}}_{k=0} + \bar{\mathbf{B}} \Delta \mathbf{u}. \tag{4.22 revisited}
\end{aligned}$$

### C.2.3 Part III

Recall the cost function which includes all the future states for a given horizon period as

$$J = \min_{\Delta \mathbf{u}_k} \frac{1}{2} \tilde{\mathbf{x}}^\top \bar{\mathbf{V}} \tilde{\mathbf{x}} - \mathbf{r}^\top \bar{\mathbf{T}} \tilde{\mathbf{x}} + \frac{1}{2} \Delta \mathbf{u}^\top \bar{\mathbf{W}} \Delta \mathbf{u} + c_1. \tag{4.21 revisited}$$

Then, by substituting for  $\tilde{\mathbf{x}}$  as in (4.22), we obtain

$$\begin{aligned}
J &= \frac{1}{2} (\bar{\mathbf{B}} \Delta \mathbf{u} + \bar{\mathbf{A}} \tilde{\mathbf{x}}_k)^\top \bar{\mathbf{V}} (\bar{\mathbf{B}} \Delta \mathbf{u} + \bar{\mathbf{A}} \tilde{\mathbf{x}}_k) - \mathbf{r}^\top \bar{\mathbf{T}} (\bar{\mathbf{B}} \Delta \mathbf{u} + \bar{\mathbf{A}} \tilde{\mathbf{x}}_k) + \frac{1}{2} \Delta \mathbf{u}^\top \bar{\mathbf{W}} \Delta \mathbf{u} + c_1, \\
&= \frac{1}{2} \Delta \mathbf{u}^\top \bar{\mathbf{B}}^\top \bar{\mathbf{V}} \bar{\mathbf{B}} \Delta \mathbf{u} + \frac{1}{2} \Delta \mathbf{u}^\top \bar{\mathbf{B}}^\top \bar{\mathbf{V}} \bar{\mathbf{A}} \tilde{\mathbf{x}}_k + \frac{1}{2} \tilde{\mathbf{x}}_k^\top \bar{\mathbf{A}}^\top \bar{\mathbf{V}} \bar{\mathbf{B}} \Delta \mathbf{u} + \frac{1}{2} \tilde{\mathbf{x}}_k^\top \bar{\mathbf{A}}^\top \bar{\mathbf{V}} \bar{\mathbf{A}} \tilde{\mathbf{x}}_k \\
&\quad - \mathbf{r}^\top \bar{\mathbf{T}} \bar{\mathbf{B}} \Delta \mathbf{u} - \mathbf{r}^\top \bar{\mathbf{T}} \bar{\mathbf{A}} \tilde{\mathbf{x}}_k + \frac{1}{2} \Delta \mathbf{u}^\top \bar{\mathbf{W}} \Delta \mathbf{u} + c_1, \\
&= \frac{1}{2} \Delta \mathbf{u}^\top \bar{\mathbf{B}}^\top \bar{\mathbf{V}} \bar{\mathbf{B}} \Delta \mathbf{u} + \frac{1}{2} (\tilde{\mathbf{x}}_k^\top \bar{\mathbf{A}}^\top \bar{\mathbf{V}} \bar{\mathbf{B}} \Delta \mathbf{u})^\top + \frac{1}{2} \tilde{\mathbf{x}}_k^\top \bar{\mathbf{A}}^\top \bar{\mathbf{V}} \bar{\mathbf{B}} \Delta \mathbf{u} + \frac{1}{2} \tilde{\mathbf{x}}_k^\top \bar{\mathbf{A}}^\top \bar{\mathbf{V}} \bar{\mathbf{A}} \tilde{\mathbf{x}}_k \\
&\quad - \mathbf{r}^\top \bar{\mathbf{T}} \bar{\mathbf{B}} \Delta \mathbf{u} - \mathbf{r}^\top \bar{\mathbf{T}} \bar{\mathbf{A}} \tilde{\mathbf{x}}_k + \frac{1}{2} \Delta \mathbf{u}^\top \bar{\mathbf{W}} \Delta \mathbf{u} + c_1,
\end{aligned}$$

Since the term  $\tilde{\mathbf{x}}_k^\top \overline{\mathbf{A}}^\top \overline{\mathbf{V}} \mathbf{B} \Delta \mathbf{u} \in \mathbb{R}^{1 \times 1}$  which is scalar,

$$J = \frac{1}{2} \Delta \mathbf{u}^\top \overline{\mathbf{B}}^\top \overline{\mathbf{V}} \mathbf{B} \Delta \mathbf{u} + \tilde{\mathbf{x}}_k^\top \overline{\mathbf{A}}^\top \overline{\mathbf{V}} \mathbf{B} \Delta \mathbf{u} - \mathbf{r}^\top \overline{\mathbf{T}} \mathbf{B} \Delta \mathbf{u} + \frac{1}{2} \Delta \mathbf{u}^\top \overline{\mathbf{W}} \Delta \mathbf{u} + c_1 + c_2, \quad (\text{C.6})$$

where

$$c_2 = \frac{1}{2} \tilde{\mathbf{x}}_k^\top \overline{\mathbf{A}}^\top \overline{\mathbf{V}} \mathbf{A} \tilde{\mathbf{x}}_k - \mathbf{r}^\top \overline{\mathbf{T}} \mathbf{A} \tilde{\mathbf{x}}_k.$$

Rearranging (C.6) and collecting like terms,

$$J = \frac{1}{2} \Delta \mathbf{u}^\top \left[ \overline{\mathbf{B}}^\top \overline{\mathbf{V}} \mathbf{B} + \overline{\mathbf{W}} \right] \Delta \mathbf{u} + \begin{bmatrix} \tilde{\mathbf{x}}_k^\top & \mathbf{r}^\top \end{bmatrix} \begin{bmatrix} \overline{\mathbf{A}}^\top \overline{\mathbf{V}} \mathbf{B} \\ -\overline{\mathbf{T}} \mathbf{B} \end{bmatrix} \Delta \mathbf{u} + c_1 + c_2. \quad (4.23 \text{ revisited})$$

## Appendix D

# Row Reduction using Gaussian Elimination

Given the matrix  $\mathbf{M}$  in (4.7) as follows:

$$\mathbf{M} = \begin{bmatrix} -\frac{\sqrt{3}}{2}k_t & \frac{\sqrt{3}}{2}k_t & 0 & 0 & 0 & 0 \\ \frac{1}{2}k_t & \frac{1}{2}k_t & -k_t & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_t & -k_t & -k_t \\ -\frac{\sqrt{3}}{2}k_d & \frac{\sqrt{3}}{2}k_d & 0 & -\frac{\sqrt{3}}{2}k_tl_0 & \frac{\sqrt{3}}{2}k_tl_0 & 0 \\ \frac{1}{2}k_d & \frac{1}{2}k_d & -k_d & \frac{1}{2}k_tl_0 & \frac{1}{2}k_tl_0 & -k_tl_0 \\ k_tl_0 & k_tl_0 & k_tl_0 & -k_d & -k_d & -k_d \end{bmatrix}.$$

It can be reduced to its row echelon form (upper triangular matrix) denoted by  $\mathbf{M}_{\text{REC}}$  by performing row operations. Note that at each step, the row operations to be performed next are stated on the right-hand side of the matrix, and the resultant matrix is given afterwards.

$$\begin{bmatrix} \textcircled{-\frac{\sqrt{3}}{2}k_t} & \frac{\sqrt{3}}{2}k_t & 0 & 0 & 0 & 0 \\ \frac{1}{2}k_t & \frac{1}{2}k_t & -k_t & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_t & -k_t & -k_t \\ -\frac{\sqrt{3}}{2}k_d & \frac{\sqrt{3}}{2}k_d & 0 & -\frac{\sqrt{3}}{2}k_tl_0 & \frac{\sqrt{3}}{2}k_tl_0 & 0 \\ \frac{1}{2}k_d & \frac{1}{2}k_d & -k_d & \frac{1}{2}k_tl_0 & \frac{1}{2}k_tl_0 & -k_tl_0 \\ k_tl_0 & k_tl_0 & k_tl_0 & -k_d & -k_d & -k_d \end{bmatrix} \begin{matrix} R2 + \frac{1}{\sqrt{3}}R1 \\ R4 - \frac{k_d}{k_t}R1 \\ R5 + \frac{k_d}{\sqrt{3}k_t}R1 \\ R6 + \frac{2l_0}{\sqrt{3}}R1 \end{matrix},$$

$$\begin{aligned}
&\Rightarrow \begin{bmatrix} -\frac{\sqrt{3}}{2}k_t & \frac{\sqrt{3}}{2}k_t & 0 & 0 & 0 & 0 \\ 0 & \textcircled{k_t} & -k_t & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_t & -k_t & -k_t \\ 0 & 0 & 0 & -\frac{\sqrt{3}}{2}k_tl_0 & \frac{\sqrt{3}}{2}k_tl_0 & 0 \\ 0 & k_d & -k_d & \frac{1}{2}k_tl_0 & \frac{1}{2}k_tl_0 & -k_tl_0 \\ 0 & 2k_tl_0 & k_tl_0 & -k_d & -k_d & -k_d \end{bmatrix} \begin{matrix} \\ R5 - \frac{k_d}{k_t}R2 \\ R6 - 2l_0R2 \end{matrix}, \\
&\Rightarrow \begin{bmatrix} -\frac{\sqrt{3}}{2}k_t & \frac{\sqrt{3}}{2}k_t & 0 & 0 & 0 & 0 \\ 0 & k_t & -k_t & 0 & 0 & 0 \\ 0 & 0 & \textcircled{0} & -k_t & -k_t & -k_t \\ 0 & 0 & 0 & -\frac{\sqrt{3}}{2}k_tl_0 & \frac{\sqrt{3}}{2}k_tl_0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}k_tl_0 & \frac{1}{2}k_tl_0 & -k_tl_0 \\ 0 & 0 & 3k_tl_0 & -k_d & -k_d & -k_d \end{bmatrix}.
\end{aligned}$$

Interchanging rows 3 and 6 yields,

$$\begin{aligned}
&\begin{bmatrix} -\frac{\sqrt{3}}{2}k_t & \frac{\sqrt{3}}{2}k_t & 0 & 0 & 0 & 0 \\ 0 & k_t & -k_t & 0 & 0 & 0 \\ 0 & 0 & \textcircled{3k_tl_0} & -k_d & -k_d & -k_d \\ 0 & 0 & 0 & -\frac{\sqrt{3}}{2}k_tl_0 & \frac{\sqrt{3}}{2}k_tl_0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}k_tl_0 & \frac{1}{2}k_tl_0 & -k_tl_0 \\ 0 & 0 & 0 & -k_t & -k_t & -k_t \end{bmatrix}, \\
&\Rightarrow \begin{bmatrix} -\frac{\sqrt{3}}{2}k_t & \frac{\sqrt{3}}{2}k_t & 0 & 0 & 0 & 0 \\ 0 & k_t & -k_t & 0 & 0 & 0 \\ 0 & 0 & 3k_tl_0 & -k_d & -k_d & -k_d \\ 0 & 0 & 0 & \textcircled{-\frac{\sqrt{3}}{2}k_tl_0} & \frac{\sqrt{3}}{2}k_tl_0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}k_tl_0 & \frac{1}{2}k_tl_0 & -k_tl_0 \\ 0 & 0 & 0 & -k_t & -k_t & -k_t \end{bmatrix} \begin{matrix} \\ \\ R5 + \frac{1}{\sqrt{3}}R4 \\ R6 - \frac{2}{\sqrt{3}l_0}R4 \end{matrix},
\end{aligned}$$

$$\begin{aligned}
&\Rightarrow \begin{bmatrix} -\frac{\sqrt{3}}{2}k_t & \frac{\sqrt{3}}{2}k_t & 0 & 0 & 0 & 0 \\ 0 & k_t & -k_t & 0 & 0 & 0 \\ 0 & 0 & 3k_tl_0 & -k_d & -k_d & -k_d \\ 0 & 0 & 0 & -\frac{\sqrt{3}}{2}k_tl_0 & \frac{\sqrt{3}}{2}k_tl_0 & 0 \\ 0 & 0 & 0 & 0 & \textcircled{k_tl_0} & -k_tl_0 \\ 0 & 0 & 0 & 0 & -2k_t & -k_t \end{bmatrix}^{R6+\frac{2}{l_0}R5}, \\
&\Rightarrow \begin{bmatrix} -\frac{\sqrt{3}}{2}k_t & \frac{\sqrt{3}}{2}k_t & 0 & 0 & 0 & 0 \\ 0 & k_t & -k_t & 0 & 0 & 0 \\ 0 & 0 & 3k_tl_0 & -k_d & -k_d & -k_d \\ 0 & 0 & 0 & -\frac{\sqrt{3}}{2}k_tl_0 & \frac{\sqrt{3}}{2}k_tl_0 & 0 \\ 0 & 0 & 0 & 0 & k_tl_0 & -k_tl_0 \\ 0 & 0 & 0 & 0 & 0 & -3k_t \end{bmatrix} = \mathbf{M}_{\text{REC}}.
\end{aligned}$$

# Appendix E

## Mixer Implementation in Pixhawk

### E.1 Airframe Configuration file

```
#!/bin/sh
#
# @name Mt_Tri_Exp
#
# @type Tricopter Y-
# @class Copter
#
# @output MAIN1 motor 1
# @output MAIN1 motor 2
# @output MAIN1 motor 3
# @output AUX1 yaw servo on motor 1
# @output AUX2 yaw servo on motor 2
# @output AUX3 yaw servo on motor 3
#
# @maintainer DA <dabara2004@yahoo.com>

sh /etc/init.d/rc.mc_defaults
```

```
if [ $AUTOCONF = yes ]
then
param set THR_MDL_FAC 0.7
param set CBRK_USB_CHK 197848
param set MPC_XY_CRUISE 3
param set MPC_XY_VEL_MAX 10
param set LNDMC_ALT_MAX 3
param set RTL_RETURN_ALT 30
param set COM_DISARM_LAND 5
param set COM_DISARM_PRFLT 15
param set SDLOG_PROFILE 11
param set EKF2_HGT_MODE 1
param set EKF2_AID_MASK 1

fi

# Configure this as Tricopter
set MAV_TYPE 15

# Set MAIN mixer / Activate outputs
set MIXER mt_tri_y_yaw-
set PWM_OUT 123

# Set AUX mixer / Activate outputs
set MIXER_AUX mt_tri_y_yaw-
set PWM_AUX_OUT 123
```



## E.2 Multi-tri Geometry file

This file defines the geometry of the multi-tilt tricopter following the PX4 framework.

```
# Tri Y

[info]
key = "3ym"
description = "Experimental Tricopter"

[rotor_default]
axis      = [0.0, 0.0, -1.0]
Ct        = 1.0
Cm        = 0.0
direction = "CW"

[[rotors]]
name      = "front_right"
position  = [0.1650, 0.2858, 0.0]

[[rotors]]
name      = "front_left"
position  = [0.1650, -0.2858, 0.0]

[[rotors]]
name      = "rear"
position  = [-0.33, 0.0, 0.0]
```

### E.3 MAIN Mixer file - `mt_tri_y_yaw-.main.mix`

This file defines the mixer for the MAIN outputs of the Pixhawk where the motors are connected.

```
# Multi-rotor tilt tricopter Main Outputs MIXER
# <Tricopter> <roll scale> <pitch scale> <yaw scale> <idle speed>
#
# Motors
R: 3ym 10000 10000 10000 0
```

### E.4 AUX Mixer file - `mt_tri_y_yaw-.aux.mix`

This file defines the AUX outputs of the Pixhawk where the servos are connected. This was done for ease of wiring. The servos may also be connected to the MAIN output depending on choice.

```
# Multi-rotor tilt tricopter Aux Mixer
# Yaw servos x3 +Output ==> -Yaw Vehicle rotation
#
# Manual passthrough for servos 1, 2 using one
# RC channel via RC_MAP_AUX1
#
# Manual passthrough for servos 1, 2, 3 using
# another RC channel via RC_MAP_AUX2
#
# Servo 1
M: 3
S: 0 2 -10000 -10000 0 -10000 10000
S: 3 5 10000 10000 0 -10000 10000
S: 3 6 -10000 -10000 0 -10000 10000
# Servo 2
M: 3
S: 0 2 -10000 -10000 0 -10000 10000
```

S: 3 5 -10000 -10000 0 -10000 10000

S: 3 6 -10000 -10000 0 -10000 10000

# Servo 3

M: 2

S: 0 2 -10000 -10000 0 -10000 10000

S: 3 6 10000 10000 0 -10000 10000