

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

Leonardo Borges Farconi

**Comparative Analysis of Robust Fault-Tolerant
Controllers Applied to Multirotor Autonomous Aerial
Vehicles**

São Carlos

2019

Leonardo Borges Farçoni

**Comparative Analysis of Robust Fault-Tolerant
Controllers Applied to Multirotor Autonomous Aerial
Vehicles**

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, para obtenção do título de Mestre em Ciências - Programa de Pós-Graduação em Engenharia Elétrica.

Área de concentração: Sistemas Dinâmicos

Supervisor: Prof. Dr. Marco Henrique Terra

**São Carlos
2019**

FOLHA DE JULGAMENTO

Candidato: Engenheiro **LEONARDO BORGES FARÇONI**.

Título da dissertação: "Análise comparativa de controladores robustos tolerantes à falhas aplicados à veículos aéreos autônomos multirotóres".

Data da defesa: 05/09/2019.

Comissão Julgadora:

Prof. Titular **Marco Henrique Terra**
(Orientador)
(Escola de Engenharia de São Carlos/EESC)

Resultado:

Aprovado

Dr. **André Marcorin de Oliveira**
(Pós-Doutorando/FAPESP)

Aprovado

Dr. **Luiz Eduardo Nunes de Almeida**
(Avibras Divisão Aérea e Naval)

Aprovado

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:
Prof. Associado **Ivan Nunes da Silva**

Presidente da Comissão de Pós-Graduação:
Prof. Titular **Murilo Araujo Romero**

ACKNOWLEDGEMENTS

Nenhum trabalho de qualidade, por maior que seja o esforço do indivíduo, é finalizado sem a colaboração de diversas entidades mesmo que de maneira indireta. Sendo assim, gostaria aqui de agradecer a todos aqueles que me auxiliaram direta ou indiretamente no cumprimento desta fase acadêmica da minha vida.

Agradeço primeiramente aos meus pais Marcos e Regina que desde sempre me deram o suporte emocional e financeiro para conquistar os meus desafios, abrindo mão de luxos e de ter-me por perto muitas vezes para que eu pudesse correr atrás dos meus sonhos.

Agradeço grandemente aos meus amigos Jéssica Bohn, Nícolas Rosa e Rafael Lang por me auxiliarem na montagem e testes do drone, assim como na companhia que mantém o trabalho sempre mais agradável. Agradeço também ao Lucas Pietro, que num pequeno gesto me ajudou a garantir minha concentração nas longas noites de escrita da dissertação.

Agradeço a empresa Avibras Indústria Aeroespacial, principalmente a meu supervisor Heitor Carlésimo e meu gerente Luiz Eduardo, pelo apoio no desenvolvimento do meu trabalho. Apoio na forma de liberação de horas para desenvolvimento, como também apoio financeiro para publicação de artigo e finalmente apoio com infraestrutura para continuação da parte prática do trabalho.

Agradeço ao Laboratório de Dinâmica da Escola de Engenharia de São Carlos, principalmente ao Prof. Marcelo Trindade, que me permitiu utilizar o cluster do laboratório, e ao técnico Leandro Pedrassolli, que me deu apoio durante todo o trabalho para utilização ininterrupta do recurso. Sem este recurso, este trabalho não seria possível. Agradeço ao apoio financeiro do MCT/CNPq/FAPEMIG Instituto Nacional de Ciência e Tecnologia em Estruturas Inteligentes em Engenharia (Projeto 574001/2008-5).

Agradeço à USP e à Escola de Engenharia de São Carlos, por me fornecerem uma das melhores infraestruturas e pessoal qualificado do país para me desenvolver como profissional. Agradeço ao CNpq, Capes, Fapesp e INSac por fornecerem patrocínio em forma de bolsa, verba para compra de material, verba para publicação de artigo, e infraestrutura para desenvolvimento do projeto, principalmente no âmbito dos projetos 2014/50851-0 (Fapesp) e 465755/2014-3 (CNPq).

Finalmente, agradeço ao meu orientador Prof. Marco Terra por confiar em mim durante todo o trabalho, me garantindo liberdade na proposição e execução do projeto, assim como dando apoio sempre que necessário. Agradeço também ao Prof. Roberto Inoue por todo o apoio durante o projeto e na revisão de artigo.

“May you live everyday of your life.”

Jonathan Swift

ABSTRACT

FARÇONI, L. **Análise Comparativa de Controladores Robustos Tolerantes à Falhas Aplicados à Veículos Aéreos Autônomos Multirotores.** 2019. 224p. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2019.

A quantidade de veículos terrestres - mais de 1 bilhão destes no mundo - tem saturado as infraestruturas viárias. Por serem controlados por motoristas, estão sujeitos a falhas decorrentes da inerente condição humana. Ambos fatores têm causado perdas econômicas e sociais crescentes. O uso de veículos aéreos autônomos com capacidade para decolagem e pouso verticais tem se apresentado como alternativa na busca da redução desses prejuízos. Não só para o transporte de pessoas, mas principalmente para o transporte de cargas. Da mesma forma, a automatização da indústria agropecuária, filmográfica, de vigilância, entre outras, vêm crescendo de maneira a incorporar sistemas autônomos aéreos em seus processos. Em todos estes casos, considerando a convivência destes sistemas com ambientes urbanos ou até mesmo o custo destes sistemas, faz-se necessária a adoção de técnicas de controle que garantam o funcionamento seguro e ininterrupto em situações de falha total de atuadores ou variações paramétricas do sistema. Este trabalho propõe o estudo aprofundado da robustez e tolerância à falha de cinco tipos básicos de controladores que se mostraram mais promissores para o controle tolerante à falhas em multirotores com redundância em hardware. É analisada também a diferença entre abordagens passivas e ativas, dando ênfase na influência que variações nos algoritmos de detecção de falha causam na estabilidade do sistema. A comparação é feita quantitativamente através de uma análise de sensibilidade dos controladores à diversas variações paramétricas e falhas após uma otimização com mínima subjetividade dos parâmetros dos controladores. Outras contribuições são feitas para a alocação de controle e para pequenas melhorias nos métodos utilizados. Os resultados são apresentados graficamente e através de tabelas de maneira a facilitar a comparação entre as abordagens. Sobressaem-se os métodos que não levam em consideração o modelo do sistema, assim como os métodos ativos, que são capazes de considerar uma gama maior de falhas.

Keywords: Multirotores; Controle tolerante à falhas; Veículo aéreo autônomo; Transporte de cargas; Análise de sensibilidade; Robustez; Simulação.

ABSTRACT

FARÇONI, L. **Comparative Analysis of Robust Fault-Tolerant Controllers Applied to Multirotor Autonomous Aerial Vehicles.** 2019. 224p. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2019.

More than 1 billion terrestrial vehicles have saturated infrastructure currently available in the world. Being controlled by human drivers, they are subject to faults inherent from human nature. Both infrastructure and driver factors have caused crescent social and economic losses. Autonomous aerial vehicles able to vertically take-off and land have been presented as an alternative to reduce those losses. Not only would that be beneficial for people transportation, but mainly for cargo transportation. In the same sense, other economy sectors such as farming, filming, vigilance and others have demonstrated interest in incorporating autonomous aerial vehicles to their businesses. In every case, coexistence of these systems with urban environments and their inherent cost require adoption of control techniques that guarantee uninterrupted and safe operation of such systems in case of total actuator failures or system parameter variations. This work proposes a thorough study of robustness and fault-tolerance of five basic controller types that showed to be promising in fault-tolerant control for hardware redundant multirotor systems. Difference between Passive and Active approaches is also analysed, emphasizing the influence that variations in the fault detection algorithms cause in system stability. Comparison is executed quantitatively through a sensitivity analysis to diverse parametric variations and faults after a optimization with minimal subjectivity of controller parameters. Other contributions are made to control allocation and small variations to improve controller robustness of some methods presented here. Results are presented graphically and by using tables to facilitate comparison between controllers. Methods that do not consider system model provide better results, as also active methods, which are capable of considering a larger pool of faults.

Keywords: Multirotor; Fault-tolerant control; Aerial autonomous vehicle; Cargo transportation; Sensitivity analysis; Robustness; Simulation.

LIST OF FIGURES

Figure 1 – EHang passenger AAV prototype	32
Figure 2 – Suggested structure for an over-actuated system. Reproduced from (JOHANSEN; FOSSEN, 2013).	34
Figure 3 – Generic multirotor and reference frames representation.	45
Figure 4 – Cascaded multirotor control architecture.	51
Figure 5 – 3D CAD representation of Model 2.	82
Figure 6 – Fitted lift coefficient for Model 2 rotors.	87
Figure 7 – Fitted drag coefficient for Model 2 rotors.	87
Figure 8 – Lift coefficient for Model 1 rotors.	88
Figure 9 – Drag coefficient for Model 1 rotors.	88
Figure 10 – Motor dynamic response for model 2 considering efficiency ξ_i	89
Figure 11 – Motor dynamic response for model 2 considering efficiency only on the parameter κ_t	90
Figure 12 – Histogram of the motor efficiency estimation error for the simulated FDD.	91
Figure 13 – Rotor architectures for cases 1 to 3, from left to right, respectively.	95
Figure 14 – Comparative study flow	101
Figure 15 – Clustering of comparison results	102
Figure 16 – Control allocation trajectory comparison for Model 2 modified.	111
Figure 17 – Control allocation rotor speeds comparison for Model 2 modified.	112
Figure 18 – Control allocation trajectory comparison for Model 2.	113
Figure 19 – Control allocation rotor speeds comparison for Model 2.	113
Figure 20 – Trajectory for Case 1. The same applies to Cases 2 and 3.	114
Figure 21 – Rotor speeds allocation for nonsymmetric ennearotor.	114
Figure 22 – Rotor speeds allocation for $W_a = 0$ in Case 2.	115
Figure 23 – Rotor speeds allocation for $W_a = 1.2 \cdot 10^{11}$ in Case 2.	116
Figure 24 – Attitude comparison for when $W_a = 0$ and $W_a = 1.2 \cdot 10^{11}$ in Case 2.	116
Figure 25 – 3D trajectory for Case 2 simulations.	117
Figure 26 – Control allocation for Case 3.	117
Figure 27 – Best individual evolution per generation for passive controllers.	118
Figure 28 – Best individual evolution per generation for active controllers.	119
Figure 29 – Histogram for factor X_1 used in Monte-carlo simulations.	119
Figure 30 – Histogram for factor X_2 used in Monte-carlo simulations.	120
Figure 31 – Histogram for factor X_3 used in Monte-carlo simulations.	121
Figure 32 – Histogram for factor X_4 used in Monte-carlo simulations.	121
Figure 33 – Histogram for factor X_5 used in Monte-carlo simulations.	121
Figure 34 – Histogram for factor X_6 used in Monte-carlo simulations.	122

Figure 35 – Histogram for factor X_7 used in Monte-carlo simulations.	122
Figure 36 – Factor-independent robustness metrics for best passive cases for Model 2.	126
Figure 37 – Total effect indices for simulation success in best passive cases for Model 2.	126
Figure 38 – Total effect indices for position error in best passive cases for Model 2.	127
Figure 39 – Total effect indices for RMS power in best passive cases for Model 2.	127
Figure 40 – Performance metrics for best passive cases for Model 2.	128
Figure 41 – Simulation success variation to FDD hit rate decrease.	131
Figure 42 – Simulation success variation to FDD delay increase.	131
Figure 43 – Factor-independent robustness metrics for best active cases for Model 2.	132
Figure 44 – Total effect indices for simulation success in best active cases for Model 2.	133
Figure 45 – Total effect indices for position error in best active cases for Model 2.	133
Figure 46 – Total effect indices for RMS power in best active cases for Model 2.	134
Figure 47 – Performance metrics for best active cases for Model 2.	134
Figure 48 – Drone parts.	157
Figure 49 – Assembling (1)	157
Figure 50 – Assembling (2)	158
Figure 51 – Assembling (3)	158
Figure 52 – Drone completed	159
Figure 53 – Flight test	159
Figure 54 – Manufacturer characterization of motor for Model 2 construction and simulation.	165
Figure 55 – Position error variation to FDD hit rate decrease.	227
Figure 56 – RMS Power variation to FDD delay increase.	227
Figure 57 – Position error variation to FDD hit rate decrease.	228
Figure 58 – RMS Power variation to FDD delay increase.	228

LIST OF TABLES

Table 1 – Robust Linear Quadratic Regulator	65
Table 2 – Robust Linear Quadratic Regulator - Always-forwards calculation	66
Table 3 – Robust Regulator for MLSU	71
Table 4 – Robust Regulator for MLSU considering only forwards calculation	72
Table 5 – Robust Regulator for MLSUM	74
Table 6 – Robust Regulator for MLSUM considering only forwards calculation	75
Table 7 – Model 1 parameters	81
Table 8 – Model 2 parameters	83
Table 9 – Multirotor parameters for Case 1	94
Table 10 – Multirotor parameters for Case 2	95
Table 11 – Multirotor parameters for Case 3	96
Table 12 – Chromosome composition	97
Table 13 – Genetic algorithm parameters	97
Table 14 – Fitness function simulation parameters	99
Table 15 – Optimization cases for passive controllers	100
Table 16 – Optimization cases for active controllers	100
Table 17 – Rotor failures set Z	106
Table 18 – Parameters X_f	106
Table 19 – Canonical rotor orientations	111
Table 20 – Summary of best fitness optimization	120
Table 21 – Factor-independent robustness metrics for Model 1 Passive Cases	123
Table 22 – Performance metrics for Model 1 Passive Cases	123
Table 23 – Factor-independent robustness metrics for Model 2 Passive Cases	124
Table 24 – System stability robustness metrics for Model 2 Passive Cases	124
Table 25 – Position error robustness metrics for Model 2 Passive Cases	124
Table 26 – RMS Power robustness metrics for Model 2 Passive Cases	125
Table 27 – Performance metrics for Model 2 Passive Cases	125
Table 28 – Factor-independent robustness metrics for Model 1 Active Cases	125
Table 29 – Performance metrics for Model 1 Active Cases	128
Table 30 – Factor-independent robustness metrics for Model 2 Active Cases	129
Table 31 – System stability robustness metrics for Model 2 Active Cases	129
Table 32 – Position error robustness metrics for Model 2 Active Cases	129
Table 33 – RMS Power robustness metrics for Model 2 Active Cases	130
Table 34 – Performance metrics for Model 2 Active Cases	130
Table 35 – Model 2 robustness to FDD variation	130
Table 36 – System stability robustness metrics for Model 1 Passive Cases	225

Table 37 – Position error robustness metrics for Model 1 Passive Cases	225
Table 38 – RMS Power robustness metrics for Model 1 Passive Cases	225
Table 39 – System stability robustness metrics for Model 1 Active Cases	226
Table 40 – Position error robustness metrics for Model 1 Active Cases	226
Table 41 – RMS Power robustness metrics for Model 1 Active Cases	226

LIST OF ABBREVIATIONS AND ACRONYMS

AAV	Autonomous Aerial Vehicles
AFTC	Active Fault-Tolerant Control(ler)
ASDB	Additive-State-Decomposition Dynamic Inversion
AUT	Aerial Urban Transport(ation)
AFT-LQR	Active Fault-Tolerant R-LQR
AFT-LQR-P	Active Fault-Tolerant R-LQR with PIDD following
AR-LQR	Active R-LQR
AM-LQR	Mode-dependent (Active) R-LQR for discrete-time Markovian jump linear systems subject to uncertainties and uncertain probability transition matrix
ASMC	SOSMC Active for virtual torque control
ASMC-D	SOSMC Active Direct
ASMC-P	SOSMC Active with PIDD following
CA	Control Allocation
CAD	Computer-Aided Design
CG	Center of Gravity
CM	Center of Mass
MLSU	Discrete-time Markovian Jump Linear System Subject to Uncertainties and uncertain probability transition matrix
MLSUM	Discrete-time Markovian Jump Linear System Subject to Uncertainties, Unknown Mode and uncertain probability transition matrix
FTC	Fault-Tolerant Control
FP	Factors Priorization
FT	Fault-Tolerant (Tolerance)
FDD	Fault Detection and Diagnosis

FT-LQR	Fault-Tolerant R-LQR
FT-LQR-P	Fault-Tolerant R-LQR with PIDD following
GA	Genetic Algorithm
INSac	Instituto Nacional de Sistemas Autônomos Cooperativos
IAE	Integral Absolute Error
ISE	Integral Squared Error
LASI	Laboratório de Sistemas Inteligentes
LPV	Linear Parameter-Varying
LMI	Linear Matrix Inequality
LQR	Linear Quadratic Regulator
LSMJ	Linear System Subject to Markovian Jumps
MIMO	Multiple-Input Multiple-Output
MRAC	Model-Reference Adaptive Control(ler)
MRAC-D	Model-Reference Adaptive Control(ler) Direct
MRAC	Model-Reference Adaptive Control(ler) with PIDD following
NDI	Non-Linear Dynamic Inversion
NSCA	Null-Space based Control Allocation
ODE	Ordinary Differential Equation(s)
PD	Proportional-Derivative
PFTC	Passive Fault-Tolerant Control(ler)
PI	Pseudo-Inverse or Proportional Integral
PID	Proportional Integral Derivative
PIDD	Proportional Integral Derivative and Second Derivative
PM-LQR	Mode-independent (Passive) R-LQR for discrete-time Markovian jump linear systems subject to uncertainties, unknown mode and unknown probability transition matrix
PSO	Particle Swarm Optimization

R-LQR	Robust LQR
RMS	Root Mean Square
RPI	Redistributed Pseudo-Inverse
SMC	Sliding-Mode Control(ler)
SOSMC	Second-Order Sliding-Mode Control(ler)
SMC-D	SOSMC Direct
SMC-P	SOSMC with PIDD following
SSMC	SOSMC for virtual torque control
TSH	Training Signal Hedging
UA	Unmanned Aircraft
USP	University of São Paulo
VTOL	Vertical Take-Off and Landing

LIST OF SYMBOLS

Symbols presented here are used throughout the text with the same meaning unless otherwise noted. For each controller section in Chapter 3, symbols were used according to the reference notation and not necessarily with this author's notation, unless otherwise noted.

$A = \{\hat{x}, \hat{y}, \hat{z}\}$ Inertial reference frame axis.

$B = \{\hat{i}, \hat{j}, \hat{k}\}$ Body-fixed reference frame axis.

\mathbf{r}_i Rotor i position in relation to B .

\mathbf{n}_i Rotor i orientation in relation to B .

ω_i Rotor speed in relation to \mathbf{n}_i .

q quaternion attitude representation of frame B .

Q Transformation matrix from B to A .

Q^{-1} Transformation matrix from A to B .

m Multirotor mass.

\mathbf{a} Multirotor linear acceleration.

\mathbf{g} Gravitational acceleration.

μ_B Matrix of aerodynamic body drag coefficients in relation to B .

\mathbf{v} Multirotor linear velocity.

F_i Rotor i thrust.

N Total number of rotors.

C_{Li} Rotor i lift coefficient.

C_{Di} Rotor i drag coefficient.

\mathbf{P} Multirotor position.

$\boldsymbol{\omega}_B$ Multirotor angular velocity.

I_{cg} Multirotor inertia tensor.

I_i Rotor i inertia in relation to \mathbf{n}_i .

κ_T	Motor torque constant.
i_i	Motor i instantaneous current.
v_{LLi}	Motor i line-to-line input voltage.
κ_v	Motor back-EMF constant.
R_i	Motor i line-to-line resistance.
\mathbf{T}^d	Desired force for body position control.
Γ_B^d	Desired torque for body attitude control.
p_i^d	Desired multirotor position for axis i .
\dot{p}_i^d	Desired multirotor velocity for axis i .
\ddot{p}_i^d	Desired multirotor acceleration for axis i .
$C = \{\hat{l}, \hat{m}, \hat{n}\}$	Frame that represents the desired multirotor attitude to execute a trajectory at.
q_ψ	Quaternion that represents attitude of C.
\mathbf{T}	Force that the multirotor can perform.
q_{ref}	Attitude reference to perform a \mathbf{T}^d .
M_f	Rotor force matrix that relates body force and rotor speeds.
Ω	Vector of squared rotor speeds.
Ω'	Vector of saturated squared rotor speeds.
q_e	Quaternion attitude error.
\mathbf{q}_e	Vector part of quaternion error.
M_t	Rotor torque matrix that relates body torque and rotor speeds.
ω^d	Vector of desired rotor speeds.
λ_i	Motor i health.
$\lambda_{i,fdd}$	Motor i health diagnosed by the FDD algorithm.
λ_{Ii}	Propeller i health.
$\lambda_{Ii,fdd}$	Propeller i health diagnosed by the FDD.

λ_i	Motor i health.
Λ_{fdd}	Matrix representation of rotor failures.
ω^{max}	Maximum rotor i speed.
ω^{min}	Minimum rotor i speed.
v_{LLi}^{max}	Maximum rotor i input voltage.
δt_{sim}	Simulation sampling period.
δt	Control loop period.
ω_B^d	Desired angular velocity.
$\dot{\omega}_B^d$	Desired angular acceleration.
ξ_i	Motor i efficiency.
$\xi_{i,fdd}$	Motor i efficiency diagnosed by the FDD algorithm.
ξ_{Pi}	Propeller i efficiency.
$\xi_{Pi,fdd}$	Propeller i efficiency diagnosed by the FDD.
$\boldsymbol{\omega}$	Vector of rotor speeds (not to be confused with ω_B).
\mathbf{r}_{pl}	Payload relative position to the CG.
m_{pl}	Payload mass.
I_{pl}	Payload inertia matrix (not to be confused with inertia tensor).
\mathbf{r}_{cg}	CG displacement in relation to the old CG position caused by the payload mass.
\mathbf{r}_{pl}^*	Payload position in relation to the CG after payload consideration.
δt_{traj}	Trajectory duration for a simulation.
t_i	Failure occurrence time.
fit_{sim}	Fitness function for one simulated trajectory.
p_{sim}	Simulation success time percentage.
$e_{p,RMS}$	Position RMS error for one simulated trajectory.
$e_{att,RMS}$	Attitude RMS error for one simulated trajectory.

P_{RMS}	RMS Power for one simulated trajectory.
$V(Y)$	Variance of Y.
$E(Y)$	Expected value of Y.
X_f	System parameter or factor to be changed in Monte-Carlo simulations.
S_f	Sensitivity index of the system to factor X_f .
S_{Tf}	Total effect index of the system to factor X_f .
S_r	Success rate.
J_c	Computational cost.

CONTENTS

1	INTRODUCTION	31
1.1	Fault-tolerant control	32
1.2	Recursive robust optimal regulator	36
1.3	Architecture-independent model, attitude planning and control allocation	37
1.4	Robustness analysis	38
1.5	Objectives	42
1.6	Dissertation structure	42
2	MULTIROTOR MODEL	45
2.1	Quaternion representation	45
2.2	Translation dynamics	47
2.3	Rotation dynamics	47
2.4	Quaternion rates	49
2.5	Motor dynamics	49
2.6	System states	50
3	CONTROL ARCHITECTURES	51
3.1	Trajectory planning	51
3.2	Position controller	52
3.3	Attitude planning	53
3.4	Attitude control	56
3.4.1	PID	56
3.4.2	SOSMC	57
3.4.2.1	SOSMC for virtual torque control (SSMC)	58
3.4.2.2	SOSMC Active for virtual torque control (ASMC)	58
3.4.2.3	SOSMC with PID following (SMC-P)	59
3.4.2.4	SOSMC Active with PID following (ASMC-P)	59
3.4.2.5	SOSMC Direct (SMC-D)	60
3.4.2.6	SOSMC Active Direct (ASMC-D)	61
3.4.3	Adaptive	61
3.4.3.1	Adaptive virtual torque control (MRAC)	63
3.4.3.2	Adaptive control with PID following (MRAC-P)	63
3.4.3.3	Adaptive Direct control (MRAC-D)	64
3.4.4	R-LQR	64
3.4.4.1	R-LQR virtual torque control (R-LQR)	65

3.4.4.2	R-LQR Active virtual torque control (AR-LQR)	66
3.4.4.3	R-LQR virtual torque control with rotor failures (FT-LQR)	67
3.4.4.4	R-LQR Active virtual torque control with rotor failures (AFT-LQR)	67
3.4.4.5	R-LQR with rotor failures and PIDD following (FT-LQR-P)	68
3.4.4.6	R-LQR Active with rotor failures and PIDD following (AFT-LQR-P)	68
3.4.5	Robust Regulator for Discrete-time Markovian Jump Linear Systems	68
3.4.5.1	Mode-dependent (AM-LQR)	69
3.4.6	Mode-independent (PM-LQR)	72
3.5	Control Allocation	75
3.5.1	Pseudo-Inverse	76
3.5.1.1	Active Pseudo-Inverse	76
3.5.2	Redistributed Pseudo-Inverse	76
3.5.2.1	Active Redistributed Pseudo-Inverse	77
3.5.3	Null-space based control allocation	77
3.5.3.1	Active Null-space based control allocation	79
4	MATERIALS AND METHODS	81
4.1	Simulation framework	83
4.1.1	Simulation solver	85
4.1.2	Control loop delay	85
4.1.3	Desired angular speeds and accelerations	86
4.1.4	Drag and lift coefficients	86
4.1.5	Fault simulation	87
4.1.6	Fault Detection and Diagnosis	90
4.1.7	Payload	91
4.1.8	Disturbance	92
4.1.9	Pseudo-Codes	92
4.2	Control allocation comparison	93
4.3	Null-space based control allocation	94
4.4	Controller optimization	96
4.4.1	GA parameters	97
4.4.2	Fitness function	97
4.4.3	Initial population	99
4.4.4	Optimization cases	99
4.5	Controller evaluation	101
4.5.1	Variance-based robustness analysis	101
4.5.2	Variance and expected value estimation	103
4.5.3	Monte-Carlo Samples	105
4.5.4	Robustness metrics	107
4.5.5	Performance metrics	108

4.5.6	FDD robustness evaluation	109
5	RESULTS	111
5.1	Control allocation comparison	111
5.2	Null-space based control allocation	112
5.2.1	Case 1: Nonsymmetric ennearotor	112
5.2.2	Case 2: Fully-actuated octarotor	114
5.2.2.1	$W_m = 1$ and $W_a = 0$	115
5.2.2.2	$W_m = 1$ and $W_a = 1.2 \cdot 10^{11}$	115
5.2.3	Case 3: Hexarotor of differently-sized propellers	115
5.3	Controller optimization	117
5.4	Controller evaluation	118
5.4.1	Samples distribution	118
5.4.2	Passive Cases	118
5.4.2.1	Model 1	122
5.4.2.2	Model 2	122
5.4.2.3	Model 2 best cases	122
5.4.3	Active Cases	123
5.4.3.1	Model 1	123
5.4.3.2	Model 2	129
5.4.3.3	Model 2 FDD robustness	129
5.4.3.4	Model 2 best cases	130
5.4.4	Discussion	132
5.4.4.1	Difference between models	132
5.4.4.2	SOSMC controllers	135
5.4.4.3	R-LQR controllers	137
5.4.4.4	Adaptive controllers	138
5.4.4.5	Comparison between best cases	139
5.4.4.6	Markovian Robust Regulators	142
6	CONCLUSION AND FUTURE WORKS	143
	BIBLIOGRAPHY	145
	APPENDIX	155
	APPENDIX A – LARGE-SIZE COAXIAL OCTACOPTER CONSTRUCTION	157

APPENDIX B – MULTICOPTER SIMULATION SCRIPT EXAM- PLE	161
APPENDIX C – MODEL 2 MOTOR CHARACTERIZATION TA- BLE	165
APPENDIX D – PSEUDO-CODES	167
APPENDIX E – BEST INDIVIDUALS AFTER CONTROLLER OP- TIMIZATIONS	181
E.1 Case 1.1.1	181
E.2 Case 1.1.2.1	181
E.3 Case 1.1.2.2	182
E.4 Case 1.1.2.3	182
E.5 Case 1.1.3.1	183
E.6 Case 1.1.3.2	184
E.7 Case 1.1.3.3	185
E.8 Case 1.1.4	187
E.9 Case 1.1.5.1	190
E.10 Case 1.1.5.2	190
E.11 Case 1.1.5.3	192
E.12 Case 1.2.1	193
E.13 Case 1.2.2.1	194
E.14 Case 1.2.2.2	194
E.15 Case 1.2.2.3	195
E.16 Case 1.2.3.1	195
E.17 Case 1.2.3.2	196
E.18 Case 1.2.3.3	197
E.19 Case 1.2.4	197
E.20 Case 1.2.5.1	200
E.21 Case 1.2.5.2	201
E.22 Case 1.2.5.3	202
E.23 Case 2.1.1	204
E.24 Case 2.1.2.1	204
E.25 Case 2.1.2.2	205
E.26 Case 2.1.2.3	206
E.27 Case 2.1.3.1	206
E.28 Case 2.1.3.2	207
E.29 Case 2.1.3.3	208
E.30 Case 2.1.4	210

E.31	Case 2.2.1	214
E.32	Case 2.2.2.1	214
E.33	Case 2.2.2.2	215
E.34	Case 2.2.2.3	215
E.35	Case 2.2.3.1	216
E.36	Case 2.2.3.2	217
E.37	Case 2.2.3.3	218
E.38	Case 2.2.4	220

APPENDIX F – SENSITIVITY INDICES FOR MODEL 1 PASSIVE AND ACTIVE CASES	225
--	------------

APPENDIX G – FDD ROBUSTNESS GRAPHS FOR MODEL 2	227
---	------------

1 INTRODUCTION

Automobiles presence in streets around the world have increased greatly in the last decades. There are estimates of more than 1.2 billion of them in the world ([GREEN CAR REPORTS, 2014](#)) and more than 101 million only in Brazil ([DENATRAN, 2019](#)). It is easy to see infrastructure development is unable to cope with vehicles number increase, making rush hour an event not limited to big cities anymore ([INRIX, 2019](#)).

In the year of 2016, americans spent an average of 225 hours commuting to work ([UNITED STATES CENSUS BUREAU, 2016](#)). The same locomotion difficulties made the average São Paulo citizen to spend an average of 1 hour and 44 minutes commuting ([FECOMERCIO SP, 2015](#)). For an Indian in Mumbai, this time gets close to 1 hour and 30 minutes ([TIMES OF INDIA, 2008](#)). In both cases, approximately 10 weeks of work are spent in transit by year. This means less time with family and leisure, more spent on fuel, less national productivity and an increase in population stress ([AL, 2012](#)).

In parallel to inefficiency statistics, the large number of vehicles in the streets associated with the human error factor brings a high level of accidents, causing large social and financial losses. Each year in the world, 1 million and 350 thousand people die in traffic accidents, while 50 million survive with injuries ([WORLD HEALTH ORGANIZATION, 2018](#)). In the year of 2014 alone, costs with accidents in Brazilian roads reached approximately R\$ 40 billion ([IPEA, 2015](#)). Of all these accidents, approximately 94 % are caused by human error ([NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION, 2015](#)).

Aerial Urban Transport (AUT), by using Vertical Take-Off and Landing Vehicles (VTOL), appears as an alternative for the urban mobility problems presented above. In the same way buildings allowed people to use land spaces more efficiently, AUT will use the tridimensional space in cities to relieve congestion on urban roads. But AUT alone is not enough. Considering the human factor to be the most probably cause of accidents, even more for a whole new environment yet to be fully adapted, it becomes necessary the use of Autonomous Aerial Vehicles (AAV). These are capable of performing take-offs, landings and routes autonomously so as to reduce human error.

Consider current car, bus and train systems. All them squeeze people in a limited number of routes from A to B in such a way that only one interruption is capable of limiting access to hundreds of people. AAVs, on the other side, are not limited to one pathway, but are capable of autonomously reducing congestion caused by lack of roads ([ANTCLIFF; GOODRICH; MOORE, 2016](#)). For this same reason, development of infrastructure for AAVs present huge economic advantages over roads, rails, bridges and tunnels. Current



Figure 1: EHang passenger AAV prototype

structures like rooftops, parking lots and empty spaces alongside roads and streets could potentially become vertical ports for AAVs. This could reduce the need for complex and expensive structures. Considering also the development of Autonomous Air Traffic Control Systems ([LOZANO, 2015](#)), the exclusion of the human error and the use of fault-tolerant systems, the social and economic cost reductions will make the use of AAVs an economic viable choice for governments around the world.

Even though this scenario seems to be far from reality, many companies around the world are developing VTOL vehicles for urban mobility. Attention should be given to Ehang ([WEISS; NICOLA, 2019](#)) - Figure 1, Volocopter ([KELLEHER, 2019](#)) and Lilium ([HAWKINS, 2019](#)), among others. Also believing in this technology, the giant Uber released a plan to start operations of AUT in its shared transport services ([UBER ELEVATION, 2016](#)), for what it will use help from Embraer to develop an aircraft for that ([GLOBO, 2017](#)). The same arguments and discussion applies to the use of AAVs for cargo transportation and heavy lifting. That is why similar initiatives are rising worldwide. Some of them are companies like SKYF ([BLAIN, 2017](#)), Boeing ([BOEING, 2019](#)) and Aerones ([AERONES, 2019](#)), among others.

1.1 Fault-tolerant control

Aerial Autonomous Vehicle Systems encompass many uncertainties which vary from atmospheric influences to component degradation and payload change. Not limited to these factors, these systems are subject to rotor and sensor failures, which can lead to catastrophic events. Considering these facts and considering that these vehicles will have

to have extreme urban mobility, the use of rotor-redundant multirotor vehicles, like coaxial octacopters, appears to be an advantageous approach. It becomes mandatory, therefore, the study of controllers capable of automatically maintaining stability and performance for such systems subject not only to uncertainties, but also to total failures of critical components such as rotors. This type of controller is called Fault-Tolerant Robust Control or only Fault-Tolerant Control (FTC) (ZHANG; JIANG, 2008).

Despite its application to diverse systems like nuclear plants (CHUNG; CHANG, 1986), chemical industries (PRAKASH; PATWARDHAN; NARASIMHAN, 2002), space systems (YEN, 1994) and more lately autonomous terrestrial vehicles (ISERMANN; SCHWARZ; STOLZL, 2002), development of FTCs was greatly motivated by aircraft development in the 70's. More specifically, fatal crashes of flights Delta 1080 (April 12th, 1977) and AA 191 (May 25th, 1979) showcased the need for fault-tolerant systems (MONTOYA et al., 1983).

Generally, fault-tolerant controllers can be classified in Passive and Active (ZHANG; JIANG, 2008)(QI et al., 2013). The first, which will be referenced as PFTC (Passive Fault-Tolerant Controller), is composed by a fixed control structure which is robust to a specific class of failures. The second, called AFTC (Active Fault-Tolerant Controller), reacts actively to component failures by reconfiguring control laws to maintain stability and performance. In general, active controllers are capable of dealing with a more severe range of failures (SANTOS et al., 2015)(MERHEB; BATEMAN; NOURA, 2015). Some reconfigurable controller types used with this intention are Linear Quadratic (VEILLETTE, 1995), Smart (ICHTEV, 2003), LMIs (LIAO; WANG; YANG, 2003) and SMCs (HESS; WELLS, 2003), among others. Hybrid systems, on the other hand, present great advantages (ZHANG; JIANG, 2008). According to (ZHANG; JIANG, 2008), for a controller to be used as AFTC, the following requirements must be satisfied:

- Controller reconfiguration must be carried in real-time;
- Controller designing must be automatic or with low human interaction or few trial-and-error attempts;
- The method shall provide a solution even if not optimal.

Simply stating, an AFTC needs a fault detection and diagnosis system (FDD) and generally is followed by a control allocation system (CA).

FDDs are sub-systems responsible for detecting a failure occurrence, locating the failure in the system and estimating its type and magnitude (QI et al., 2013) in the quickest possible way (ZHANG; JIANG, 2008). Even though this sub-system offers valuable information for system operation, maintenance and development, it is considered a major challenge since the time for fault diagnostic is limited and any diagnostic error

may lead to problems for the reconfigurable controller. FDD will not be focused in this work. Provisions will be made to consider the existence of such systems, as presented on Section 4.1.6. More information can be obtained in (QI et al., 2013) and (ZHANG; JIANG, 2008).

As previously stated, rotor-redundant systems can be used for obtaining failure-robustness. In these types of systems, control can be divided in three levels, as presented by Figure 2: Body control by using a virtual vector τ_c of forces and moments; force and moment distribution for each actuator; and individual actuator control to guarantee execution of the demanded forces and moments (JOHANSEN; FOSSEN, 2013). The second level responsible for forces and moments distribution to actuators is called Control Allocation. This system modularity facilitates design in some cases, but may not exist in others. According to (JOHANSEN; FOSSEN, 2013), the CA problem can be divided in linear and non-linear actuator modelling. The first consider a model of the type

$$\tau_c = Bu, \quad (1.1)$$

while the second is represented by

$$\tau_c = h(u, x, t), \quad (1.2)$$

where τ_c is the vector of virtual forces and moments, u is the actuator input, $B \in \mathbb{R}^{m \times p}$ is the effective control matrix and h is a non-linear function. This work will deal with the linear case only.

CA methods can also be classified in Unconstrained and Constrained. The first does not consider maximum or minimum actuator responses, providing simpler solutions by means of generalized inverses or decomposition in singular values (OPPENHEIMER; DOMAN; BOLENDER, 2010). The latter considers actuator limitations and does not have analytical solutions, providing solutions by means of Redistributed Pseudo-Inverse (RPI) (SHI et al., 2010), Daisy Chaining (BUFFINGTON; ENNS, 1996), Direct Allocation (DURHAM, 1993), Error minimization by Linear Programming (BODSON, 2002), Active Set Quadratic Programming (HARKEGARD, 2004), and Adaptive Control Allocation (CASAVOLA; GARONE, 2010). More details can be obtained in (JOHANSEN; FOSSEN, 2013).

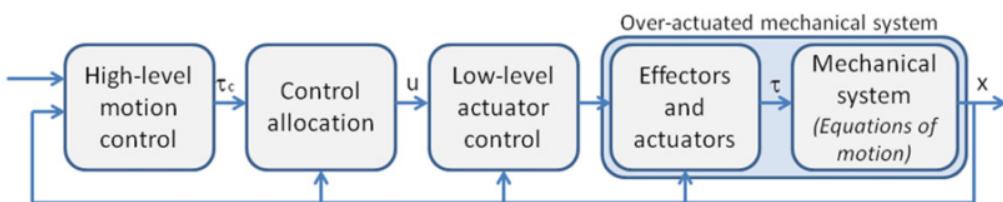


Figure 2: Suggested structure for an over-actuated system. Reproduced from (JOHANSEN; FOSSEN, 2013).

Even though study of fault-tolerant systems presents to be crucial, this is a relatively new area, which does not have standards when it comes to concepts, methods, and even terminologies (ZHANG; JIANG, 2008). Moreover, few are the works when it comes to application to control of multirotor aircrafts. (MARKS; WHIDBORNE; YAMAMOTO, 2012) and (SAIED et al., 2015) use the classic PD controller in octacopters. (SAIED et al., 2015) focuses in detection of errors with residuals generated by Sliding Mode observers so as to compensate failures with offline computed strategies for each failure case. This strategy is able to tolerate complete failure of only one rotor and is less robust by limiting actuator of the rotors left. (MARKS; WHIDBORNE; YAMAMOTO, 2012), on the other hand, uses CA by RPI, assuming an FDD delay of 0.5 s, which gives more satisfactory results with total rotor failures up to four rotors.

(GUANGXUN; QUAN; KAI-YUAN, 2013), (FALCONI; MARVAKOV; HOLZAPFEL, 2016), (FALCONI; HEISE; HOLZAPFEL, 2015) and (FALCONI; SCHATZ; HOLZAPFEL, 2016) present passive controllers based in Additive-State-Decomposition Dynamic inversion (ASDB), Incremental Backstepping and Non-Linear Dynamic Inversion (NDI). As presented by (FALCONI; MARVAKOV; HOLZAPFEL, 2016) and (ÖZBEK; ÖNKOL; EFE, 2016), model inversion methods similar to these generally report degraded performances for larger modelling errors or different payload configurations. This is reflected in the fact that these four works can deal with only one partial rotor failure. (FALCONI; HOLZAPFEL, 2016), on the other side, even though making use of Backstepping, presents an adaptive control allocation method that extends the Backstepping controller robustness to one total rotor failure.

The majority of works in this multirotor field use Sliding Mode Control (SMC), since this type of controller posses great robustness to faults, parameter variations and disturbances (MERHEB; BATEMAN; NOURA, 2015)(SAIED et al., 2016a). Attention should be given to (ALWI; EDWARDS, 2013),(ALWI; HAMAYUN; EDWARDS, 2014),(ALWI; EDWARDS, 2015),(MERHEB; BATEMAN; NOURA, 2015) and (SAIED et al., 2016a), which, in the best case, can control octacopters with up to four rotor failures, followed by 20% of loss in 3 rotors more and 10% of loss in an eighth rotor. In the worst case, controller exhibit 50% fault in four motors and a total failure in a fifth motor. It should be noted, however, that if one does not consider real actuator scenarios with actuator limitations, the robustness conclusion may be false or limited. (ALWI; EDWARDS, 2013) presents an SMC structure which writes the problem in a Linear Parameter Varying (LPV) format and uses of fault-weighted Pseudo-Inverse for CA, which can be considered as an active approach. (ALWI; HAMAYUN; EDWARDS, 2014) makes no use of FDD algorithms by considering rotor failures as control disturbances and by using the Integral Sliding Mode Control method, which increases system robustness forcing it to go to the sliding surface. This technique uses a nominal feedback matrix, becoming a hybrid architecture with an LQR. (ALWI; EDWARDS, 2015) extends the disturbance concept of (ALWI; HAMAYUN;

EDWARDS, 2014) to (ALWI; EDWARDS, 2013) and compares both active and passive approaches. (MERHEB; BATEMAN; NOURA, 2015) and (SAIED et al., 2016a) use the Second-Order Sliding Mode Control (SOSMC), which enhances accuracy and robustness, guarantees convergence in finite time and reduces chattering, according to (MERHEB; BATEMAN; NOURA, 2015) and (ÖZBEK; ÖNKOL; EFE, 2016). (MERHEB; BATEMAN; NOURA, 2015) also uses a non-linear sliding surface for the SOSMC and compares active and passive methods as in (ALWI; EDWARDS, 2015), presenting SOSMC-based observer which compares to an offline test table to obtain an FDD algorithm. (SAIED et al., 2016a) proposes the use of Super-Twisting Algorithm, differently of (MERHEB; BATEMAN; NOURA, 2015), which gives less satisfactory results.

A promising method, which was used for fixed-wing aircraft, is presented in (SCHWAGER; ANNASWAMY; LAVRETSKY, 2005). (SCHWAGER; ANNASWAMY; LAVRETSKY, 2005) proposes a direct adaptive controller, based in Model Reference Adaptive Control (IOANNOU; SUN, 2012) and Training Signal Hedging (TSH) (KARASSON; ANNASWAMY, 1994), which deals not only with actuator faults, disturbances and parameter variations, but also with actuator limitations. Results can be applied to multirotor aircrafts considering system modelling resemblance.

A detailed survey on AFTCs and SMCs can be obtained in (ZHANG; JIANG, 2008) and (SPURGEON, 2014), respectively. More details are available in each reference.

1.2 Recursive robust optimal regulator

Many works present optimal controllers for fault-tolerant systems as summarized by (ZHANG; JIANG, 2008). (SATICI; POONAWALA; SPONG, 2013), (PANOMRATTA-NARUG; HIGUCHI; MORA-CAMINO, 2013), (LIU et al., 2013) and (NAVABI; MIRZAEI, 2017) present examples of optimal controllers for multirotors subject to small uncertainties. When dealing with large or total rotor failures for redundant multirotor systems, like hexa or octacopters, however, there are no contributions that make use of recursive optimal controllers, to the extent of this author's knowledge.

(CERRI, 2009) presents an optimal recursive robust regulator. The solution is equal to the traditional LQR when uncertainties are not considered. This regulator considers the existence of uncertainties linked to both the state and input matrices of a discrete linear system, which may be time variant. (LEAO, 2015) used this regulator to control a quadcopter subject to mass and moment of inertia uncertainties, comparing the solution to other approaches. Given the modelling approach used by (LEAO, 2015), however, rotor uncertainties were not considered. In this sense, it is important to analyse this controller when rotor uncertainties are considered, which makes it a potential method for fault-tolerant control in multirotors. More details can be obtained in Section 3.4.4.

Rotor failures can be considered as uncertainties in the system model in some cases. They change system behaviour drastically, however. Considering this, it is possible that modelling rotor failures as uncertainties may not be enough, reducing controller performance greatly. If one consider rotor failures to be random events during the multirotor's life time, one could classify a multirotor subject to rotor failures as a Linear System subject to Markovian Jumps (LSMJ) (COSTA; MARQUES; FRAGOSO, 2005) (SIQUEIRA; TERRA; BERGERMAN, 2011). This type of system can be modelled differently for each operation mode. In this sense, each rotor failure could be modelled as a different operation mode. (CERRI, 2013) developed a recursive optimal controller for this kind of problem, considering not only different modes of operation, but also uncertainties for each mode, as long as it is possible to tell which mode the system is operating at. (BORTOLIN; TERRA, 2015) extended this concept for when it is not possible to know the current system mode of operation. Considering this is a recent and not so explored solution, and considering its potential in dealing with faulty systems, it is necessary to study this class of controllers applied to multirotor fault-tolerant control. More details are presented on Section 3.4.5.

1.3 Architecture-independent model, attitude planning and control allocation

Quadrotors provide great maneuverability and low-cost maintenance, compared to helicopters (LEE; LEOK; MCCLAMROCH, 2010). Diverse architectures have been created to overcome quadrotors when it comes to safety (SAIED et al., 2016b), efficiency (BLAIN, 2017) and functionality (BRESCIANINI; D'ANDREA, 2016). However, in general, each new architecture proposition comes with its own dynamic model, attitude planning and control allocation scheme (JOHANSEN; FOSSEN, 2013). By attitude planning, we call the calculation of the angles which the multirotor has to be tilted to so as to execute a desired movement. By control allocation, we call the distribution of effort to each actuator so as to compose body force and moment vectors (JOHANSEN; FOSSEN, 2013), like presented on Section 1.1. This first modelling study characterizes a tedious task which could be avoided.

Many works similar to (IRELAND; VARGAS; ANDERSON, 2015) and (LEE; LEOK; MCCLAMROCH, 2010) assume a common underactuated multirotor architecture in which all rotors face upwards, generating thrust all in the same upwards direction \hat{k} . The attitude planning procedure, in this case, consists primarily of finding roll and pitch angles at which the thrust in this direction can be equal to the position control effort. This assumption, however, is limited in the sense that it does not support fully-actuated structures similar to the one presented on Section 5.2.2, among others.

This work presents an attitude planning and control allocation algorithm not based on a specific multirotor architecture. The method allows for an arbitrary attitude reference to be defined and considers a preferred direction of impulse that is not necessarily equal

to \hat{k} . Reference ([BENZAID; MANSOURI, 2016](#)) argues to provide a generic model but limits the multirotor architecture to “+” or “ \times ” configurations. On ([CAI et al., 2017](#)) a more generalized model is presented. It even includes blade-flapping effects, but fails to address attitude planning and control allocation. This latter issue could be considered more important than the modelling itself, since it is used directly in controlling rotor speeds. Reference ([MICHIELETTTO; RYLL; FRANCHI, 2017](#)) was successful in accomplishing both modelling, attitude planning and control allocation in a generalized manner for statically hoverable multirotors, applied to an hexarotor robustness analysis. The author presents a model based on first principles and rigid-body dynamics; an attitude planning based on rotation matrices; and a control allocation based on the non-orthogonality between the null-space of the control moment input matrix and the row space of the control force input matrix. Despite this great contribution, the author’s solution is not final, considering that it presents an infinite number of solutions to the problem based on an arbitrary vector choice by the reader (vector b_2 in ([MICHIELETTTO; RYLL; FRANCHI, 2017](#))).

In this paper, we present three main contributions regarding modelling, attitude planning and control allocation: In the first, it improves the generalized approaches described above by utilizing quaternion representation. In the second, it extends the attitude planning and the control allocation algorithms from ([MICHIELETTTO; RYLL; FRANCHI, 2017](#)) by providing a method to choose a final solution based on a cost function between maneuverability and attitude reference following ability. And finally, in the third, it analyses the algorithm capabilities with different multirotor architectures: a Nonsymmetric ennearotor (9 rotors), an octarotor with four outside-directed propellers and an hexarotor with two large and four small propellers.

These generalized algorithms for statically hoverable multirotors can be of use when investigating different multirotor architectures’ characteristics and control algorithms for them, jump-starting to essential analysis without spending effort on attitude planning and control allocation. They are also useful for allocation in rotor-fail situations, which is the case of this work, and even in architecture-changing multirotors. Moreover, quaternions have no limitations regarding singularities, in opposition to what occurs with Euler Angles at *Gimbal Lock*. They are also proven to be faster, more efficient and more stable than Euler Angles, that rely on trigonometric functions ([ALAIMO et al., 2013](#)), ([ALAIMO et al., 2014](#)).

1.4 Robustness analysis

Robustness is often treated vaguely or too qualitatively when dealing with multirotors. Not many works address rotor failures in multirotors with more than four rotors. The ones that address this issue do not have the rigor to analyse either how much a given controller can handle faults and parameter variations, or how is a given controller

robustness compared to previous suggested architectures. Moreover, subjectivity is always strongly present either in controller tuning or evaluation.

(XIONG; ZHANG, 2017) proposes a sliding mode control for position and attitude tracking for quadrotors. Controller parameters are calculated based on analytical stability analysis, which guarantees asymptotically stability. The model, however is simplified, ignoring rotor inertia. The author argues for robustness but no quantitative metrics are proposed to measure robustness. Disturbances are also not specified and only one case is simulated. (NGUYEN; HONG, 2018) presents a fault-tolerant control for quadcopters using robust adaptive sliding mode control. The author verifies stability using Lyapunov theory, proposing to handle faults and uncertainties. Different failure cases are tested but only with uncertainty in inertia varied once for one test. Moreover, no metrics are used to compare improvements, where terms like "quick compensation" and "good trajectory tracking" are used with only figures to corroborate. (NAVABI; MIRZAEI, 2017) gives a good contribution by introducing a Robust Optimal Adaptive controller for trajectory tracking with quadrotors. The work analyses stability from the analytical point of view and also suggests a particle swarm optimization for one controller case optimization, reducing subjectivity. It simulates disturbance, parameter variation and actuator limitations but with only a few specific cases. The lack of quantitative metrics weakens the robustness argument. (LEÃO; INOQUE; TERRA, 2015) is successful in introducing more than one controller for comparison. It simulates many cases with aleatory mass and inertia variations, analysing results over one quadratic error metric. This approach is good but may be not enough to give better insights considering the few number of parameter variations and only one metric used. The model is also simplified, not considering actuator response, faults and uncertainties.

(GIERNACKI; SADALLA, 2017) and (LIAO; WANG; YANG, 2003) do not deal with multirotors specifically but can also be used as examples. (GIERNACKI; SADALLA, 2017) treats about rotor control robustness for two different tuning methods. Compared to the works above, this author is successful in using standardized metrics like the IAE and ISE and analysing their variations along with parameter variations. (LIAO; WANG; YANG, 2003) on the other hand, compares robust tracking approaches for fixed wing aircraft. The author proposes a mixed method and compares it to a classical one using the same metrics in both cases, while considering gust disturbance and actuator dynamics. The number of simulation cases, however, could be improved for better comparison.

On the octacopter segment, (SAIED et al., 2015) tests a FTC method over one hovering case and a circular trajectory case. Only one motor failure is simulated. The drawbacks here are qualitative characteristic of comparisons, assuming robustness to unknown parameter variation and noise without any quantifiable argument. (YOON; CICHELLA; HOVAKIMYAN, 2016) proposes adaptive algorithms for both the attitude

control and control allocation. The author gives a great analytical synthesis, proving controller stability and considering some uncertainties in the controller definition. The control allocation estimation is not guaranteed to converge to the real value, however, even though the author argues that response in this case is better. The proposed allocation is compared with a regular allocation by means of three different failures inserted in the same simulation. Motor dynamics and time delay are considered, however, there are no quantitative metrics for comparison and no analysis of robustness to disturbance and model uncertainties even though the author vouches for them. (BRITO, 2016) does a great job synthesising theories of diverse FT controllers that could be used for rotor failure robustness in coaxial octacopters. However, only two controllers are compared in the work. The author is careful by recognizing the need for a non-subjective controller optimization by choosing a PSO to tune parameters. The final comparison, though, is based greatly on qualitative analysis.

An interesting approach is presented by (MICHIELETTO; RYLL; FRANCHI, 2017). Instead of analysing controller performance and robustness, the author investigates the multirotor architecture robustness analytically by analysing whether there are feasible rotor commands to perform flight after failures. This is made for two hexarotor architectures, showcasing differences in robustness by tilting the rotors. This approach is a good tool to help designing new aircraft.

Finally, (ÖZBEK; ÖNKOL; EFE, 2016) and (SHRAIM; AWADA; YOUNESS, 2018) are better examples of comparative analysis for quadrotors. They excel by introducing diverse metrics and comparing various control algorithms. Even though (ÖZBEK; ÖNKOL; EFE, 2016) argues that controller responses are very sensitive to the chosen controller parameters, the author uses trial-and-error for tuning with different objectives for each controller, making the comparison subjective. The author presents standardized metrics for comparison but fails when comparing computational complexity without exposing the method used for that. Moreover, there is no description of how disturbances and uncertainties were inserted in simulations besides sensor errors. Simulation trials are also very limited. (SHRAIM; AWADA; YOUNESS, 2018) on the other side, is presented as a summary of quadcopter control from system configuration to fault-tolerant control. Controller tuning tries to reduce subjectivity by iterating in a loop that varies in a previously defined space of allowable values obtained from stability analysis. This method is not explained, though. The author presents a good robustness analysis tentative but fails in the number of simulation cases and in specifying disturbances and metrics.

Considering the scenario presented above, there is much improvement to be made in robustness analysis of multirotor fault-tolerant control. Even more in the case of total actuator failures for redundant systems. (CAMBONE et al., 2005) brings the importance of this subject by stating that “Improving UA (Unmanned Aircraft) reliability is the

single most immediate and long-reaching need to ensure their success". This work is a tentative of overcoming the issues presented above by (1) trying to use representative multirotor modelling, (2) investigating diverse control architectures that have shown to be the most suitable, according to Section 1.1, (3) reducing controller tuning subjectivity by evolutive optimization algorithms, (4) executing thousands of monte-carlo simulations to vary diverse parameters and failure cases, and finally (5) using quantitative metrics to analyse performance and robustness not only in terms of stability but also in terms of individual parameter influence in control stability. The use of quantifiable and well-defined metrics is stressed here because the overconfidence of modellers may cause robustness analysis to be more mentioned than practised, resulting in incomplete understanding of system input-output relationship ([SALTELLI et al., 2004](#)).

Analytical analysis of parameter influence in control stability for all controller cases studied here would be unfeasible, to not say impossible. Therefore, a numeric method is proposed based on sensitivity analysis as presented by ([SALTELLI et al., 2004](#)). Sensitivity analysis is defined by ([SALTELLI et al., 2004](#)) as "The study of how the uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input". This work looks for a sensibility analysis with the ability to consider the inputs probability distribution; the ability to evaluate the effect of an input while all other inputs are varying; the ability to be model-independent; and the ability of treating grouped factors as if they were single factors, to simplify analysis. Even though diverse methods are available, ([SALTELLI et al., 2004](#)) argues that the methods of Variance-Based sensitivity analysis are able to cope with all the cited desirable properties.

([SALTELLI et al., 2004](#)) summarizes the use of sensitivity analysis in four different settings:

Factors prioritization: To identify the most important factor that would lead to the greatest reduction in output variance if fixed, giving a rational choice to focus either on model construction or system analysis when dealing with uncertainties.

Factors fixing: Used mostly for simplifying complex models, this setting has the objective of finding the factor that can be fixed at any value, usually its nominal value, without influencing the output variance.

Variance cutting: To select the least number of parameters that are able to reduce the output variance to a predefined value, even not knowing the values to fix the parameters at.

Factors mapping: To identify which factor is most responsible for producing realizations of the system that are acceptable or not acceptable.

This work will focus on the Factors Priorization Setting so as to analyse which factor or parameter most influence a controller robustness. More details will be presented on Section 4.5.

1.5 Objectives

The major purpose of this work is to execute a throughout investigation of robustness and performance of five different controller architectures - namely PID, SOSMC ([MERHEB; BATEMAN; NOURA, 2015](#)), Adaptive ([SCHWAGER; ANNASWAMY; LAVRETSKY, 2005](#)), Recursive Robust LQR ([CERRI, 2009](#)) and Recursive Robust LQR for Linear Systems Subject to Markovian Jumps ([BORTOLIN, 2017](#)) - in Passive and Active settings and in a redundant multirotor system subject to total rotor failures and parameter variations, making this the first work of this kind, to the extent of this author's knowledge. Hybrid architectures could also be studied and provide better results, but considering the limited time for the project, attention is given to understand advantages and drawbacks of each base architecture.

Minor objectives consist of

- Proposing minor modifications to improve controller robustness, leading to a pool of controllers to be investigated;
- Investigating FDD influence on Active controllers of the same architectures above;
- Improving multirotor architecture-independent modelling by using quaternions;
- Improving attitude planning and CA algorithms proposed by ([MICHIELETTI; RYLL; FRANCHI, 2017](#));
- Comparing basic CA methods;
- Developing structured code for multirotor simulation; and
- Constructing a medium-size octacopter for future testing of results obtained.

1.6 Dissertation structure

Chapter 1 presented a brief motivational review of some statistics and current development in the field of civil AAVs, stating the importance of the subject here studied.

Chapter 2 develops a generalized multirotor dynamic model to be used in simulations, explaining quaternion representation, translational and rotational dynamics, motor dynamics and final state space variables.

Chapter 3 composes the main text body. It starts by introducing the trajectory planning method and position controller. Attitude planning method is then presented

followed by all attitude controllers and their variations. Finally, three CA methods are presented for later comparison.

Chapter 4 explains how investigations and comparisons were carried out. Attention is given first to details regarding simulation. Next sections comprise methods of comparing CA methods, investigating Null-Space based CA, optimizing control architectures from Chapter 3 and evaluating control architectures for robustness and performance analysis.

Chapter 5 presents results for each investigation and optimization, and for the main robustness analysis, dividing results in Passive and Active controllers and for Model 1 and Model 2, as will be further explained.

Chapter 6 summarizes the main contributions of this work and points out future works and main improvements to be made.

Appendices at the end of dissertation have the purpose of not polluting the main text body. They are referenced within the text when necessary.

2 MULTIROTOR MODEL

In this chapter, we are going to obtain a generalized dynamic multirotor model. The generic multirotor can be represented as shown in Fig. 3, where the inertial reference frame A is defined by $\{\hat{x}, \hat{y}, \hat{z}\}$, the body-fixed reference frame centered at the center of gravity (CG) of a multirotor B is defined by $\{\hat{i}, \hat{j}, \hat{k}\}$, the three-dimensional position and orientation vector of the rotor i , concerning the origin of B, are given by \mathbf{r}_i and \mathbf{n}_i , respectively, and the rotor speed of rotor i in radians per second, following the right-hand rule over \mathbf{n}_i , is given by ω_i .

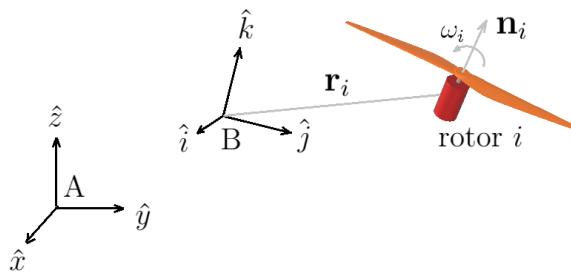


Figure 3: Generic multirotor and reference frames representation.

In order to obtain the dynamic equations that represent the multirotor of Fig. 3, in the next sections, we are going to present quaternion operations and change of basis using them, the rigid-body dynamics and the relation between body angular velocity and quaternion rates. The quaternion representation used in the multirotor model of this paper is a significant contribution compared to ([MICHELETTO; RYLL; FRANCHI, 2017](#)).

2.1 Quaternion representation

Quaternions are a mathematical representation developed by W.R. Hamilton which contains four dimensions. They can represent attitude in tri-dimensional space and are strictly related to Euler's Rotation Theorem. The latter, in a simplified explanation, states that any displacement in tri-dimensional space, in which a point remains fixed, can be represented by a rotation over an axis through this point.

Quaternion representation has advantages for not presenting limitations related to singularities, in opposition to what can happen with Euler Angles during a *Gimbal Lock*. Quaternions are proved to be faster, more efficient and more stable compared to Euler Angles, which rely on trigonometric functions ([ALAIMO et al., 2013](#)) ([ALAIMO et al., 2014](#)).

A quaternion, $q \in \mathbb{H}$, is represented by

$$q = q_1 + q_2i + q_3j + q_4k, \quad (2.1)$$

where i , j and k are imaginary numbers, with relations and algebra explained in ([HAMILTON, 1844](#)). One can represent q as a vector ([DIEBEL, 2006](#)), which can be easily related to the unitary rotation axis \mathbf{n} and rotation angle α , around \mathbf{n} , if $|q| = 1$, by

$$q = [q_1, q_2, q_3, q_4]^T \doteq [q_1, \mathbf{q}^T]^T \doteq \begin{bmatrix} \cos\left(\frac{1}{2}\alpha\right) \\ \mathbf{n} \sin\left(\frac{1}{2}\alpha\right) \end{bmatrix}. \quad (2.2)$$

Let $\mathbf{v}_A = (\mathbf{v})_A$ be a vector \mathbf{v} in \mathbb{R}^3 represented in frame A. If one makes $\mathbf{v}_A = v_1i + v_2j + v_3k$,

$$\mathbf{v}_B = q \otimes \mathbf{v}_A \otimes q^{-1}, \quad (2.3)$$

gives the same vector in relation to body reference frame B, $(\mathbf{v})_B$, where \otimes is the Hermitian product when dealing with quaternions, and q is the body attitude ([DIEBEL, 2006](#)). Following quaternion algebra ([HAMILTON, 1844](#)), ([DIEBEL, 2006](#)), the Hermitian product can be represented in vector form by

$$q \otimes p = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & q_4 & -q_3 \\ q_3 & -q_4 & q_1 & q_2 \\ q_4 & q_3 & -q_2 & q_1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}, \quad (2.4)$$

which leads to the vector representation form

$$\mathbf{v}_B = Q^{-1}\mathbf{v}_A, \quad (2.5)$$

where Q^{-1} is the transformation matrix from A to B, and $Q = (Q^{-1})^T$ is the transformation matrix from B to A, where

$$Q = \begin{pmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2q_3 - q_1q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_2q_3 + q_1q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{pmatrix}. \quad (2.6)$$

Throughout this work, $Q(q)$ will always refer to the above transformation from arbitrary frames B to A, where A is the same as B rotated by $(q)_A$. Capital letter indices represent the reference frame to which a variable is being reference to, unless otherwise noted. Shortly, $\mathbf{v}_C = (\mathbf{v})_C$, $\mathbf{v}_B = (\mathbf{v})_B$ and $\mathbf{v}_A = (\mathbf{v})_A$ refer to the same variable written in respect to frames C, B and A.

2.2 Translation dynamics

Let

$$\sum \mathbf{F}_B = m\mathbf{a}_B, \quad (2.7)$$

be the sum of all forces acting on the multirotor, where m is the multirotor's mass and \mathbf{a} is the multirotor's linear acceleration. Considering that the multirotor is subject to weight, aerodynamic body drag and the forces generated by all rotors, Equation (2.7) becomes

$$m\mathbf{g}_B - \mu_B \mathbf{v}_B + \sum_1^N F_i \mathbf{n}_i = m\mathbf{a}_B, \quad (2.8)$$

where \mathbf{g} is the gravitational acceleration, \mathbf{v} is the multirotor's linear velocity, N is the total number of rotors,

$$F_i = C_{Li} \omega_i^2 \quad (\text{POUNDS et al., 2004}) \quad (2.9)$$

is the lift of rotor i in the direction of \mathbf{n}_i , C_{Li} is the lift coefficient of rotor i and

$$\mu_B = \begin{pmatrix} \mu_{\hat{i}\hat{i}} & \mu_{\hat{i}\hat{j}} & \mu_{\hat{i}\hat{k}} \\ \mu_{\hat{j}\hat{i}} & \mu_{\hat{j}\hat{j}} & \mu_{\hat{j}\hat{k}} \\ \mu_{\hat{k}\hat{i}} & \mu_{\hat{k}\hat{j}} & \mu_{\hat{k}\hat{k}} \end{pmatrix}$$

is the constant matrix of aerodynamic body drag coefficients. μ_B was chosen constant in relation to body frame considering that drag is closely related to the body format and direction of wind in relation to it.

Rewriting Equation (2.8) in relation to frame A yields

$$\dot{\mathbf{v}}_A = \mathbf{g}_A + \left(\frac{1}{m}\right) Q \sum_1^N [C_{Li} \omega_i^2 \mathbf{n}_i] - \left(\frac{1}{m}\right) Q \mu_B Q^{-1} \mathbf{v}_A, \quad (2.10)$$

where $\dot{\mathbf{v}}_A = \mathbf{a}_A$, $\mathbf{g}_A = [0, 0, -g]^T$ and g is the absolute value of the gravity acceleration.

The multirotor position derivative is defined as

$$\dot{\mathbf{P}}_A = \mathbf{v}_A. \quad (2.11)$$

2.3 Rotation dynamics

Consider the sum of moments around the CG as (HIBBEKER, 2016)

$$\sum M_{cg} = \dot{H}_{cg}, \quad (2.12)$$

where H_{cg} is the angular quantity of movement.

Considering frame B is moving with angular velocity $\boldsymbol{\omega}$,

$$(\dot{H}_{cg})_B = (\dot{H}_{cg})_{\hat{i}\hat{j}\hat{k}} + \boldsymbol{\omega}_B \times (H_{cg})_B \quad (2.13)$$

where $(\dot{H}_{cg})_{\hat{i}\hat{j}\hat{k}} = \dot{H}_i\hat{i} + \dot{H}_j\hat{j} + \dot{H}_k\hat{k}$ is the derivative of H_{cg} measured from the origin of B and $(\dot{H}_{cg})_B$ is the total derivative of H written in reference to B.

Reference ([HIBBELE, 2016](#)) also states that

$$(H_{cg})_B = I_{cg} \cdot \boldsymbol{\omega}_B, \quad (2.14)$$

where

$$I_{cg} = \begin{pmatrix} I_{\hat{i}\hat{i}} & -I_{\hat{i}\hat{j}} & -I_{\hat{i}\hat{k}} \\ -I_{\hat{j}\hat{i}} & I_{\hat{j}\hat{j}} & -I_{\hat{j}\hat{k}} \\ -I_{\hat{k}\hat{i}} & -I_{\hat{k}\hat{j}} & I_{\hat{k}\hat{k}} \end{pmatrix}, \quad (2.15)$$

is the multirotor's inertial tensor measured from the CG, which is constant. $I_{\hat{i}\hat{i}}$, $I_{\hat{j}\hat{j}}$ and $I_{\hat{k}\hat{k}}$ are the body's main moments of inertia. $I_{\hat{i}\hat{j}} = I_{\hat{j}\hat{i}}$, $I_{\hat{i}\hat{k}} = I_{\hat{k}\hat{i}}$ and $I_{\hat{j}\hat{k}} = I_{\hat{k}\hat{j}}$ are the body's products of inertia. Therefore,

$$\left(\sum M_{cg} \right)_B = I_{cg} \cdot \dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times (I_{cg} \cdot \boldsymbol{\omega}_B). \quad (2.16)$$

The moment generated by the lift of one rotor is

$$f_i = C_{Li}\omega_i^2 (\mathbf{r}_i \times \mathbf{n}_i). \quad (2.17)$$

Applying Equation (2.12) to each rotor, and considering that the angular quantity of movement in a rotor is composed mainly by the rotation in the rotor axis direction, one can show that

$$\begin{aligned} \sum M_i &= \dot{H}_i, \\ &= (\dot{H}_i)_{\hat{i}\hat{j}\hat{k}} + \boldsymbol{\omega}_B \times (H_i)_B, \\ &= [\dot{I}_i(\omega_i \mathbf{n}_i) + I_i(\dot{\omega}_i \mathbf{n}_i + \omega_i \dot{\mathbf{n}}_i)] + \boldsymbol{\omega}_B \times (I_i \omega_i \mathbf{n}_i), \\ &= I_i \dot{\omega}_i \mathbf{n}_i + I_i \omega_i (\boldsymbol{\omega}_B \times \mathbf{n}_i), \end{aligned} \quad (2.18)$$

where $\sum M_i$ is the sum of moments in the rotor i , H_i is the angular quantity of movement of rotor i , I_i is the rotor inertia around the \mathbf{n}_i axis, $\dot{I}_i = 0$, and $\dot{\mathbf{n}}_i = 0$ because there is no blade-flapping.

The sum of moments in the rotor is composed by external torques to the rotor system, M'_i , the torque generated by the rotor, τ_i , and the drag torque

$$-C_{Di}\omega_i |\omega_i| \mathbf{n}_i, \quad (2.19)$$

where C_{Di} is the drag torque coefficient for rotor i . The use of the square of rotor speeds for lift and torque, as suggested in Equations (2.9) and (2.19), does not loses generality. One could replace this literature model by PWM percentage or control throttle. One can

assume that $\sum M_i$ is composed by torques in the \mathbf{n}_i direction and torques not in this direction. Therefore,

$$M'_i = I_i \omega_i (\boldsymbol{\omega}_B \times \mathbf{n}_i), \quad (2.20)$$

$$\tau_i = C_{Di} \omega_i |\omega_i| \mathbf{n}_i + I_i \dot{\omega}_i \mathbf{n}_i. \quad (2.21)$$

The external torques to the rotor system exert direct influence in the multirotor, while the torque generated by the rotor exert an opposite reaction torque to the multirotor's body through the rotor armature. Therefore, the moments generated by a rotor i to the multirotor's body are

$$M''_i = M'_i - \tau_i. \quad (2.22)$$

Hence, Equation (2.16) becomes

$$I_{cg} \cdot \dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times (I_{cg} \cdot \boldsymbol{\omega}_B) = \sum_1^N [f_i + M'_i - \tau_i], \quad (2.23)$$

$$\dot{\boldsymbol{\omega}}_B = I_{cg}^{-1} [(I_{cg} \boldsymbol{\omega}_B - a(t)) \times \boldsymbol{\omega}_B + b(t)], \quad (2.24)$$

where

$$a(t) = \sum_1^N I_i \omega_i \mathbf{n}_i, \quad (2.25)$$

$$b(t) = \sum_1^N [C_{Li} \omega_i^2 (\mathbf{r}_i \times \mathbf{n}_i) - (C_{Di} \omega_i |\omega_i| + I_i \dot{\omega}_i) \mathbf{n}_i]. \quad (2.26)$$

2.4 Quaternion rates

Let $(R(t))_B = q^{-1}(t) \cdot R(0) \cdot q(t)$ be a point that rotates with the rigid body. Knowing that $(\dot{R}(t))_B = \boldsymbol{\omega}_B \times (R(t))_B$, one can show that (ALAIMO et al., 2013)

$$\dot{q} = \frac{1}{2} S(q) \boldsymbol{\omega}_B, \quad (2.27)$$

where

$$S(q) = \begin{pmatrix} -q_2 & -q_3 & -q_4 \\ q_1 & -q_4 & q_3 \\ q_4 & q_1 & -q_2 \\ -q_3 & q_2 & q_1 \end{pmatrix}. \quad (2.28)$$

2.5 Motor dynamics

Brushless motors are the best choice for multirotors considering their higher efficiency, higher torque-to-weight ratio and less susceptibility to wear considering the lack of mechanical commutation. From Equation (2.21), one can obtain the scalar relation

$$\dot{\omega}_i = \frac{1}{I_i} (\tau_i - C_{Di} \omega_i |\omega_i|), \quad (2.29)$$

which describes the rotor dynamics. Reference ([GIERAS, 2010](#)) states that the brushless motor torque can be approximated by

$$\tau_i = \kappa_T i_i, \quad (2.30)$$

where κ_T is the motor torque constant and i_i is the motor instantaneous current. The current, according also to ([GIERAS, 2010](#)), can be obtained from

$$i_i = \frac{v_{LLi} - \kappa_v \omega_i}{R_i}, \quad (2.31)$$

where v_{LLi} is the motor Line-to-Line input voltage, κ_v is the motor back-EMF constant and R_i is the motor Line-to-Line electrical resistance. It is important to notice that the motor inductance was neglected due to the fact that the motor electrical response is expressively faster than the mechanical response. By considering the inductance in the simulation model, one can make simulation time a lot slower because of increased model order without expressive modelling gains.

2.6 System states

For a multirotor model without the motor modelling, the system state is comprised of \mathbf{P}_A , \mathbf{v}_A , q and $\boldsymbol{\omega}_B$ while rotor speeds ω_i , for $i = 1, \dots, N$, are the system inputs. Equations (2.11), (2.10), (2.27) and (2.24) define the state derivatives of the multirotor model.

For a multirotor model that considers the motor modelling, system state is comprised of \mathbf{P}_A , \mathbf{v}_A , q , $\boldsymbol{\omega}_B$ and $\dot{\omega}_i$, for $i = 1, \dots, N$, while motor voltage inputs v_{LLi} are the system inputs. The state derivatives are the same as above with the addition of Equation (2.29).

The use of more realistic dynamic models allows for the development of more robust controllers ([MOHAMMADI; Mohammad Shahri, 2013](#)). As model-based design tools, these models reduce development costs linked to countless test trials ([Abu Rmilah; HASSAN; Bin Mardi, 2016](#)) and help to discover development difficulties on early project stages, increasing return-on-investment ([RUFF; STEPHENS; MAHAPATRA, 2012](#)).

3 CONTROL ARCHITECTURES

Multiactuated systems, like multirotors, are dynamically cascaded (JOHANSEN; FOSSEN, 2013): a high-level control commands a vector of virtual force (\mathbf{T}^d) and torque (Γ_B^d) based on a trajectory planning; a control allocation coordinates the actuators to execute the virtual commands; and a low-level control commands each actuator individually. A few examples of this architecture can be found in (LEAO, 2015), (MAHONY; KUMAR; CORKE, 2012), (IRELAND; VARGAS; ANDERSON, 2015), (RINALDI; GARGIOLI; QUAGLIOTTI, 2014) and (BENZAID; MANSOURI, 2016).

The high-level control can be further divided on position control, attitude planning and attitude control. This happens because generally, multirotors are sub-actuated systems in such a way that it is not possible to directly control their position, but instead, it is necessary to tilt them so as to guarantee the execution of a force in the desired control direction. Figure 4 summarizes this control architecture. Some controllers, as will be seen further below and we call here "Direct", do not require a control allocation scheme.

The following sections try to explain each control phase in the sequence proposed on Figure 4. Section 3.1 presents the trajectory planning that feeds the position controller detailed on Section 3.2. Section 3.3 explains the attitude planning algorithm that comes before the attitude controller detailed on Section 3.4. Finally, the control allocation algorithms are presented on Section 3.5.

3.1 Trajectory planning

Diverse trajectory planning algorithms are available on literature, which is beyond the scope of this work. The trajectory planning algorithm used here is based on a simple method for robotic manipulators presented on (CRAIG, 2005). This method creates a

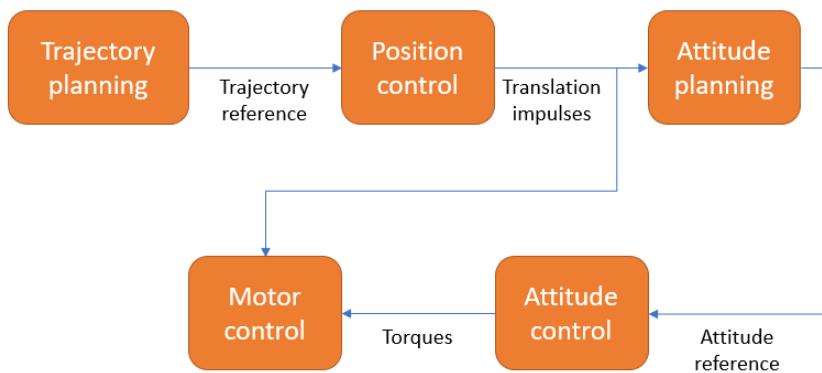


Figure 4: Cascaded multirotor control architecture.

unique and smooth trajectory, with smooth velocity curves between initial and final states. Reference (LEAO, 2015) adapted this method for a quadrotor, using it to generate independent trajectories for each Cartesian axis.

Let i be an axis, namely $i = \hat{x}, \hat{y}$ or \hat{z} . The desired position, velocity and acceleration for the moment t in i can be calculated, respectively, by

$$\begin{aligned} p_i^d(t) &= a_{i1} + a_{i2}\Delta t + a_{i3}\Delta t^2 + a_{i4}\Delta t^3 + a_{i5}\Delta t^4 + a_{i6}\Delta t^5 \\ \dot{p}_i^d(t) &= a_{a2} + 2a_{i3}\Delta t + 3a_{i4}\Delta t^2 + 4a_{i5}\Delta t^3 + 5a_{i6}\Delta t^4 \\ \ddot{p}_i^d(t) &= 2a_{i3} + 6a_{i4}\Delta t + 12a_{i5}\Delta t^2 + 20a_{i6}\Delta t^3 \end{aligned} \quad (3.1)$$

where $\Delta t = t - t_0$ is the difference between the initial and final times and where $\mathbf{a}_i = [a_{i1} \ a_{i2} \ a_{i3} \ a_{i4} \ a_{i5} \ a_{i6}]^T$ can be calculated by

$$\mathbf{a}_i = \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & (t_f - t_0) & (t_f - t_0)^2 & (t_f - t_0)^3 & (t_f - t_0)^4 & (t_f - t_0)^5 \\ 0 & 1 & 2(t_f - t_0) & 3(t_f - t_0)^2 & 4(t_f - t_0)^3 & 5(t_f - t_0)^4 \\ 0 & 0 & 2 & 6(t_f - t_0) & 12(t_f - t_0)^2 & 20(t_f - t_0)^3 \end{array} \right]^{-1} \mathbf{p}_i \quad (3.2)$$

with $\mathbf{p}_i = [p_{i0} \ \dot{p}_{i0} \ \ddot{p}_{i0} \ p_{if} \ \dot{p}_{if} \ \ddot{p}_{if}]^T$. t_0 is the initial time, at which the initial trajectory states are p_{i0}, \dot{p}_{i0} , and \ddot{p}_{i0} . Similarly, t_f is the final moment when trajectory reaches the desired final states p_{if}, \dot{p}_{if} , e \ddot{p}_{if} .

3.2 Position controller

The position controller calculates the necessary impulse, \mathbf{T}_A^d , for the multirotor to follow a trajectory reference. This is usually achieved by means of a PID controller similar to the one presented in (LUUKKONEN, 2011) and transcribed in Equation 3.3.

$$\begin{aligned} \mathbf{T}_A^d &= K_P \begin{bmatrix} p_{\hat{x}}^d(t) - P_{A\hat{x}} \\ p_{\hat{y}}^d(t) - P_{A\hat{y}} \\ p_{\hat{z}}^d(t) - P_{A\hat{z}} \end{bmatrix} + K_I \int \begin{bmatrix} p_{\hat{x}}^d(t) - P_{A\hat{x}} \\ p_{\hat{y}}^d(t) - P_{A\hat{y}} \\ p_{\hat{z}}^d(t) - P_{A\hat{z}} \end{bmatrix} \\ &+ K_D \begin{bmatrix} \dot{p}_{\hat{x}}^d(t) - v_{A\hat{x}} \\ \dot{p}_{\hat{y}}^d(t) - v_{A\hat{y}} \\ \dot{p}_{\hat{z}}^d(t) - v_{A\hat{z}} \end{bmatrix} + K_{DD} \begin{bmatrix} \ddot{p}_{\hat{x}}^d(t) - \dot{v}_{A\hat{x}} \\ \ddot{p}_{\hat{y}}^d(t) - \dot{v}_{A\hat{y}} \\ \ddot{p}_{\hat{z}}^d(t) - \dot{v}_{A\hat{z}} \end{bmatrix} - m\mathbf{g}_A, \end{aligned} \quad (3.3)$$

where $p^d(t)$, $\dot{p}^d(t)$ and $\ddot{p}^d(t)$ are the desired position, velocities and accelerations, respectively. The PIDD coefficients are

$$\begin{aligned} K_P &= \text{diag}(K_{Px}, K_{Py}, K_{Pz}), \\ K_I &= \text{diag}(K_{Ix}, K_{Iy}, K_{Iz}), \\ K_D &= \text{diag}(K_{Dx}, K_{Dy}, K_{Dz}), \\ K_{DD} &= \text{diag}(K_{DDx}, K_{DDy}, K_{DDz}). \end{aligned} \quad (3.4)$$

The second derivative terms are responsible for dealing with high-order dynamics present in multirotor architectures.

3.3 Attitude planning

The virtual torque Γ_B^d can be calculated after knowing the attitude which the actuator has to be tilted to.

Many works similar to (IRELAND; VARGAS; ANDERSON, 2015) assume a common underactuated multirotor architecture in which all rotors face upwards, generating thrust only in the \hat{k} direction. The attitude planning procedure, in this case, consists primarily of finding roll and pitch angles at which the thrust in \hat{k} direction can be equal to \mathbf{T}_A^d . (FALCONI; MARVAKOV; HOLZAPFEL, 2016), differently, defines an attitude error vector between the translation free vector and \hat{k} axis that generates, by a PD control, a torque vector for the multirotor to follow the position controller.

This work presents an attitude planning algorithm not based on a specific multirotor architecture. It uses the concept of attitude error like in (FALCONI; MARVAKOV; HOLZAPFEL, 2016), but to generate an attitude reference to be delivered to an attitude controller, like in (IRELAND; VARGAS; ANDERSON, 2015). The method allows for an attitude reference to be defined and considers a preferred direction of impulse that is not necessarily equal to \hat{k} . For common architectures, the method is similar to the ones used in (IRELAND; VARGAS; ANDERSON, 2015) or (FALCONI; MARVAKOV; HOLZAPFEL, 2016). However, it is applicable to multirotor architectures like the one in (BRESCIANINI; D'ANDREA, 2016) or in studies of diverse multirotor architectures.

Step 1 : Suppose the multirotor is capable of executing the impulse \mathbf{T}^d at an attitude q_ψ . Usually, this attitude is the hovering with specified yaw.

Step 2 : Calculate the rotor speeds necessary to generate \mathbf{T}^d at q_ψ .

Step 3 : If rotor speeds are within the permissible range, the reference attitude is q_ψ itself. If not, limit rotor speeds and recalculate the impulse that the multirotor can perform while in q_ψ .

Step 4 : The final attitude reference will be the one necessary to align the impulse vector generated on step 3 to \mathbf{T}^d .

Consider the following:

- $C \equiv \{\hat{l}, \hat{m}, \hat{n}\}$ is the body reference frame that represents the multirotor at a desired attitude to perform the movement at.
- q_ψ is the quaternion that represents the attitude of C . If q_ψ is the hover attitude at a given ψ yaw,

$$(q_\psi)_A = [\cos(\psi/2) \ 0 \ 0 \ \sin(\psi/2)]^T, \quad (3.5)$$

which gives

$$Q_{ca} = Q(q_\psi). \quad (3.6)$$

- \mathbf{T}^d is the desired translational impulse requested by the position controller.
- \mathbf{T} is the impulse that the multirotor can perform.
- q_{ref} is the final attitude reference for B .

Relating to Equation (2.10) one can define the translation impulse generated by the rotors as

$$\mathbf{T}_C \equiv \mathbf{T}_B = \sum_1^N [C_{Li} \omega_i^2 \mathbf{n}_i]. \quad (3.7)$$

One can rewrite Equation (3.7) as a linear system

$$\mathbf{T}_C = [C_{Li} \mathbf{n}_i, \dots, C_{LN} \mathbf{n}_N] \begin{bmatrix} \omega_1^2 \\ \vdots \\ \omega_N^2 \end{bmatrix} = M_f \boldsymbol{\Omega}. \quad (3.8)$$

The rotor speeds, $\boldsymbol{\Omega}$, can be calculated by means of a control allocation method for \mathbf{T}_C^d , like the ones described in Section 3.5. Here, we refer to the one in Section 3.5.3

without the allocation for the virtual torque, which means making $\boldsymbol{\Omega} = \boldsymbol{\Omega}_f$. The Moore-Penrose pseudo-inverse (BARATA; HUSSEIN, 2012) (Section 3.5.1) or the Redistributed Pseudo-Inverse (Section 3.5.2) could be less robust alternatives. Therefore,

$$\boldsymbol{\Omega} = f(\mathbf{T}_C^d) = f(Q_{ca}^{-1}\mathbf{T}_A^d), \quad (3.9)$$

where $f(\cdot)$ represents a transformation from virtual force to rotor speeds.

Rotors have a maximum and minimum allowable speed ω^{max} and $\omega^{min} \geq 0$. Hence,

$$\omega'_i = \begin{cases} \omega_i^{max}, & \text{if } \omega_i \geq \omega_i^{max} \\ \omega_i^{min}, & \text{if } \omega_i \leq \omega_i^{min} \\ \omega_i, & \text{otherwise} \end{cases}, \quad (3.10)$$

and

$$\mathbf{T}_A = Q_{ca}M_f\boldsymbol{\Omega}', \text{ for } \boldsymbol{\Omega}' = [(\omega'_1)^2 \dots (\omega'_N)^2]^T. \quad (3.11)$$

Let θ_{cb} and \mathbf{v}_{cb} be the angle and unit vector, respectively, that rotates T_A to T_A^d , or in other words, from C to B. The quaternion that represents this rotation is

$$(q_{cb})_A = \begin{bmatrix} \cos\left(\frac{\theta_{cb}}{2}\right) \\ (\mathbf{v}_{cb})_A \sin\left(\frac{\theta_{cb}}{2}\right) \end{bmatrix}, \quad (3.12)$$

where

$$\theta_{cb} = \arccos\left(\frac{\mathbf{T}_A \cdot \mathbf{T}_A^d}{\|\mathbf{T}_A\| \|\mathbf{T}_A^d\|}\right), \quad (3.13)$$

$$(\mathbf{v}_{cb})_A = \frac{\mathbf{T}_A}{\|\mathbf{T}_A\|} \times \frac{\mathbf{T}_A^d}{\|\mathbf{T}_A^d\|}. \quad (3.14)$$

Using the relation presented in Equation (2.3), one can show that

$$\begin{aligned} \mathbf{T}_B &= (q_{cb})_A \cdot \mathbf{T}_C \cdot (q_{cb}^{-1})_A, \\ \mathbf{T}_B &= (q_{cb})_A \cdot (q_\psi)_A \cdot \mathbf{T}_A \cdot (q_\psi^{-1})_A \cdot (q_{cb}^{-1})_A, \\ \mathbf{T}_B &= (q_{ref})_A \cdot \mathbf{T}_A \cdot (q_{ref}^{-1})_A, \end{aligned} \quad (3.15)$$

which is the same as saying that the composition of two rotations is the product of the quaternions that represent each rotation.

Therefore, using Equation (2.4),

$$q_{ref} = \begin{bmatrix} q_{cb1} & -q_{cb2} & -q_{cb3} & -q_{cb4} \\ q_{cb2} & q_{cb1} & q_{cb4} & -q_{cb3} \\ q_{cb3} & -q_{cb4} & q_{cb1} & q_{cb2} \\ q_{cb4} & q_{cb3} & -q_{cb2} & q_{cb1} \end{bmatrix} \begin{bmatrix} q_\psi 1 \\ q_\psi 2 \\ q_\psi 3 \\ q_\psi 4 \end{bmatrix}. \quad (3.16)$$

3.4 Attitude control

This section presents the attitude control algorithms to be compared to each other. Considering the presented on Sections 1.1, 1.2 and 1.5, five basic attitude control architectures will be compared, namely PID, SOSMC, Adaptive, R-LQR and Markovian. Each one has variations proposed here to investigate whether improvements can be made in relation to fault-tolerance.

When it comes to the use of FDD schemes, the controllers are divided on "passive" and "active". Regarding the R-LQR controllers, one modification includes the rotor failures in the model (Sections 3.4.4.3 and 3.4.4.4). Another modification tries to decrease the multirotor sensitivity to mass and linear parameter variations by including the position controller force in the attitude controller. Those cases are presented on Sections 3.4.2.3, 3.4.2.4, 3.4.3.2, 3.4.4.5 and 3.4.4.6. The last proposed modification avoids the use of control allocators and is presented on Sections 3.4.2.5, 3.4.2.6 and 3.4.3.3. In total there are 19 attitude controller architectures considering the PID architecture with "active" control allocation.

Considering the use of quaternions the attitude error can be represented by a quaternion that rotates from the current attitude to the desired one. According to (FRESK; NIKOLAKOPOULOS, 2013), this error quaternion can be obtained by

$$q_e = q_{ref} \otimes q^{-1} = \begin{bmatrix} q_{e0} \\ \mathbf{q}_e \end{bmatrix},$$

$$q_e = \begin{bmatrix} q_{ref2}q_2 + q_{ref3}q_3 + q_{ref4}q_4 + q_{ref1}q_1 \\ -q_{ref1}q_2 - q_{ref4}q_3 + q_{ref3}q_4 + q_{ref2}q_1 \\ q_{ref4}q_2 - q_{ref1}q_3 - q_{ref2}q_4 + q_{ref3}q_1 \\ -q_{ref3}q_2 + q_{ref2}q_3 - q_{ref1}q_4 + q_{ref4}q_1 \end{bmatrix}. \quad (3.17)$$

The vector part of q_e , \mathbf{q}_e , is equivalent in direction to a control angular acceleration which can bring q to q_{ref} (FRESK; NIKOLAKOPOULOS, 2013). If the reference demands a rotation angle greater than π according to Equation (2.2), the closest rotation can be obtained by reverting the rotation axis. Therefore,

$$\text{if } q_{e0} < 0, \mathbf{q}_e \leftarrow -\mathbf{q}_e \quad (3.18)$$

3.4.1 PID

The attitude PID control uses the same principle from Section 3.2 but without the second derivative (ALAIMO et al., 2014). Hence, the virtual control torque is

$$\boldsymbol{\Gamma}_B^d = I_{cg}[K_{\tau P} \mathbf{q}_e + K_{\tau I} \int \mathbf{q}_e + K_{\tau D}((\boldsymbol{\omega}_{ref})_B - \boldsymbol{\omega}_B)], \quad (3.19)$$

where

$$\begin{aligned} K_{\tau P} &= \text{diag}(K_{\tau Px}, K_{\tau Py}, K_{\tau Pz}), \\ K_{\tau I} &= \text{diag}(K_{\tau Ix}, K_{\tau Iy}, K_{\tau Iz}), \\ K_{\tau D} &= \text{diag}(K_{\tau Dx}, K_{\tau Dy}, K_{\tau Dz}) \end{aligned}$$

are the PID coefficients and ω_{ref} is an angular velocity reference for the multicopter to follow. In this case $\omega_{ref} = 0$, which yields

$$\mathbf{T}_B^d = I_{cg} \left[K_{\tau P} \mathbf{q}_e + K_{\tau I} \int \mathbf{q}_e - K_{\tau D} \boldsymbol{\omega}_B \right]. \quad (3.20)$$

3.4.2 SOSMC

Following the same notation as in ([MERHEB; BATEMAN; NOURA, 2015](#)), one can define the Second-Order Sliding Mode Controller according to the description below.

Let

$$\ddot{x} = f(x) + g(x)u(t) \quad (3.21)$$

be the state-space representation of the system to be controlled, with $x, f(x) \in \mathbb{R}^n$, $g(x) \in \mathbb{R}^{n \times m}$ and $u(t) \in \mathbb{R}^m$ the control input.

Assume a sliding surface

$$s = \dot{e} + ce, \quad (3.22)$$

where

$$e = x_d - x \quad (3.23)$$

is the tracking error for a desired state x_d and $c \in \mathbb{R}^{n \times n}$ is a diagonal matrix of positive values. s becomes zero when the system hits the surface, giving the error an exponentially decreasing behaviour of $\dot{e} = -ce$. For when the system is not in the sliding surface, it shall be brought to it by using discontinuous functions. In the Second-Order Sliding Mode Controller, the surface second derivative $\ddot{s} = \ddot{e} + c\dot{e}$ is also set to zero, which yields from

$$\ddot{e} = -c\dot{e} = \ddot{x}_d - \ddot{x} \quad (3.24)$$

the equivalent control law u_{eq} . The SOSMC control law can be obtained, therefore, by

$$u = u_{eq} + u_{dis}, \quad (3.25)$$

$$u_{eq} = g^{-1}(x) [\ddot{x}_d + c\dot{e} - f(x)], \quad (3.26)$$

$$u_{dis} = \lambda |s|^2 \text{sat}(s) + \int \alpha \text{sat}(s) dt \quad (3.27)$$

$$\text{sat}(s) = \min(\max(s, -1), 1) \quad (3.28)$$

where λ and $\alpha \in \mathbb{R}$ are positive gain vectors. ([MERHEB; BATEMAN; NOURA, 2015](#)) originally uses the *signum* function instead of the saturation function. He later points

out that the saturation function can increase motor speed smoothness. In simulations the saturation functions were necessary to provide better system convergence.

For finite-time convergence, λ and α must be chosen so as to guarantee

$$\alpha > \frac{\Phi}{G_2} \quad (3.29)$$

$$\lambda \geq \sqrt{\frac{4\Phi G_2(\alpha + \Phi)}{G_1^2 G_1(\alpha - \Phi)}} \quad (3.30)$$

where G_1 , G_2 and Φ are restrictions on the second-order derivative of s and

$$\ddot{s} = \varphi(t, x, u) + G(t, x, u)\dot{u}, \quad (3.31)$$

$$|\varphi(t, x, u)| \leq \Phi, \quad (3.32)$$

$$0 \leq G_1 \leq G(t, x, u) \leq G_2 \quad (3.33)$$

(MERHEB; BATEMAN; NOURA, 2015) presents this controller in a passive architecture, without any FDD or CA. He indicates, however, that failure robustness could be increased by using a CA method configurable through a FDD algorithm.

3.4.2.1 SOSMC for virtual torque control (SSMC)

Using the modelling notation of this work, one can apply the SOSMC for multirotor attitude control by making

$$\begin{aligned} f(x) &\leftarrow I_{cg}^{-1} [(I_{cg}\boldsymbol{\omega}_B - a(t)) \times \boldsymbol{\omega}_B] & g(x) &\leftarrow I_{cg}^{-1} & \ddot{x} &\leftarrow \dot{\boldsymbol{\omega}}_B \\ \ddot{x}_d &\leftarrow \dot{\boldsymbol{\omega}}_B^d & \dot{e} &\leftarrow \boldsymbol{\omega}_B^d - \boldsymbol{\omega}_B & e &\leftarrow \mathbf{q}_e & u &\rightarrow \boldsymbol{\Gamma}_B^d \end{aligned} \quad (3.34)$$

3.4.2.2 SOSMC Active for virtual torque control (ASMC)

When a FDD algorithm is present, the methodology of 3.4.2.1 can be slightly altered to update the controller architecture online in case of rotor failures. Let $\lambda_{Ii|fdd} \in \mathbb{R}$ | $0 \leq \lambda_{Ii|fdd} \leq 1$ to represent the health of rotor i identified by a FDD method according to Section 4.1.6, with $\lambda_{Ii|fdd} = 0$ meaning a total failure diagnose and $\lambda_{Ii|fdd} = 1$ a healthy rotor diagnose. $a(t)$ from Equation (2.25) and present in Equation (3.34), can be altered to

$$a_{fdd}(t) = \sum_1^N I_i \sqrt{\lambda_{i|fdd}} \omega_i \mathbf{n}_i, \quad (3.35)$$

resulting in the SOSMC Active control for multirotor of Equation (3.36).

$$\begin{aligned} f(x) &\leftarrow I_{cg}^{-1} [(I_{cg}\boldsymbol{\omega}_B - a_{fdd}(t)) \times \boldsymbol{\omega}_B] & g(x) &\leftarrow I_{cg}^{-1} & \ddot{x} &\leftarrow \dot{\boldsymbol{\omega}}_B \\ \ddot{x}_d &\leftarrow \dot{\boldsymbol{\omega}}_B^d & \dot{e} &\leftarrow \boldsymbol{\omega}_B^d - \boldsymbol{\omega}_B & e &\leftarrow \mathbf{q}_e & u &\rightarrow \boldsymbol{\Gamma}_B^d \end{aligned} \quad (3.36)$$

3.4.2.3 SOSMC with PIDD following (SMC-P)

The first modification includes in the error definition terms of Equation (2.8). The error considers the desired body position, obtained from simple kinematics detailed on Equations (3.37) and (3.38) over the desired position control force. Rather than using $p_i^d(t)$, the idea here is to consider linear parameters in the attitude control and to force the multirotor to follow the position controller in case of faults, instead of substituting the position controller of Section 3.2 entirely. The position controller is necessary because the attitude planning uses the force vector to obtain an attitude reference.

From kinematics, let

$$\begin{aligned}\dot{\mathbf{v}}_A^d &= \frac{\mathbf{v}_A^d(t + \delta t) - \mathbf{v}_A(t)}{\delta t} \\ \dot{\mathbf{v}}_A^d &= \frac{\mathbf{P}_A^d(t + \delta t) - 2\mathbf{P}_A(t) + \mathbf{P}_A(t - \delta t)}{\delta t^2} \\ \mathbf{P}_A^d(t + \delta t) &= \dot{\mathbf{v}}_A^d \delta t^2 + 2\mathbf{P}_A(t) - \mathbf{P}_A(t - \delta t),\end{aligned}\quad (3.37)$$

where δt is the controller loop time, from one iteration to another. Defining the position error as the difference between the desired position and current position, one can get

$$\begin{aligned}\mathbf{e}_{\mathbf{P}_B} &= \mathbf{P}_A^d(t + \delta t) - \mathbf{P}_A(t) = Q^{-1} \left[\dot{\mathbf{v}}_A^d \delta t^2 + \mathbf{P}_A(t) - \mathbf{P}_A(t - \delta t) \right] \\ \mathbf{e}_{\mathbf{P}_B} &= Q^{-1} \left[\frac{\mathbf{T}_A^d}{m} \delta t^2 + \mathbf{P}_A(t) - \mathbf{P}_A(t - \delta t) \right].\end{aligned}\quad (3.38)$$

Therefore, using the same notation as in Section 3.4.2.1, the SOSMC with PIDD can be applied to multirotor control by making

$$\begin{aligned}f(x) &\leftarrow \begin{bmatrix} I_{cg}^{-1} [(I_{cg}\boldsymbol{\omega}_B - a(t)) \times \boldsymbol{\omega}_B] \\ Q^{-1}\mathbf{g}_A - \mu_B Q^{-1}\frac{\mathbf{v}_A}{m} \end{bmatrix} & g(x) &\leftarrow \begin{bmatrix} I_{cg}^{-1} & 0_{3x3} \\ 0_{3x3} & 1_{3x3}/m \end{bmatrix} & \ddot{x} &\leftarrow \begin{bmatrix} \dot{\boldsymbol{\omega}}_B \\ \dot{\mathbf{v}}_B \end{bmatrix} \\ \ddot{x}_d &\leftarrow \begin{bmatrix} \dot{\boldsymbol{\omega}}_B^d \\ Q^{-1}\frac{\mathbf{T}_A^d}{m} \end{bmatrix} & \dot{e} &\leftarrow \begin{bmatrix} \boldsymbol{\omega}_B^d - \boldsymbol{\omega}_B \\ Q^{-1}\frac{\mathbf{T}_A^d}{m} \delta t \end{bmatrix} & e &\leftarrow \begin{bmatrix} \mathbf{q}_e \\ \mathbf{e}_{\mathbf{P}_B} \end{bmatrix} & u &\rightarrow \begin{bmatrix} \boldsymbol{\Gamma}_B^d \\ Q^{-1}\mathbf{T}_A^{d*} \end{bmatrix},\end{aligned}\quad (3.39)$$

where \mathbf{T}_A^{d*} shall substitute \mathbf{T}_A^d in next steps of control allocation.

3.4.2.4 SOSMC Active with PIDD following (ASMC-P)

Online controller updating by a FDD algorithm can be implemented here by using the exact same method as in Section 3.36. Therefore, the ASMC-P can be applied to multirotor control by making

$$\begin{aligned}f(x) &\leftarrow \begin{bmatrix} I_{cg}^{-1} [(I_{cg}\boldsymbol{\omega}_B - a_{fdd}(t)) \times \boldsymbol{\omega}_B] \\ Q^{-1}\mathbf{g}_A - \mu_B Q^{-1}\frac{\mathbf{v}_A}{m} \end{bmatrix} & g(x) &\leftarrow \begin{bmatrix} I_{cg}^{-1} & 0_{3x3} \\ 0_{3x3} & 1_{3x3}/m \end{bmatrix} & \ddot{x} &\leftarrow \begin{bmatrix} \dot{\boldsymbol{\omega}}_B \\ \dot{\mathbf{v}}_B \end{bmatrix} \\ \ddot{x}_d &\leftarrow \begin{bmatrix} \dot{\boldsymbol{\omega}}_B^d \\ Q^{-1}\frac{\mathbf{T}_A^d}{m} \end{bmatrix} & \dot{e} &\leftarrow \begin{bmatrix} \boldsymbol{\omega}_B^d - \boldsymbol{\omega}_B \\ Q^{-1}\frac{\mathbf{T}_A^d}{m} \delta t \end{bmatrix} & e &\leftarrow \begin{bmatrix} \mathbf{q}_e \\ \mathbf{e}_{\mathbf{P}_B} \end{bmatrix} & u &\rightarrow \begin{bmatrix} \boldsymbol{\Gamma}_B^d \\ Q^{-1}\mathbf{T}_A^{d*} \end{bmatrix},\end{aligned}\quad (3.40)$$

where \mathbf{T}_A^{d*} shall substitute \mathbf{T}_A^d in next steps of control allocation.

3.4.2.5 SOSMC Direct (SMC-D)

This proposed SOSMC variation changes the output u from virtual forces and torques to desired rotor speeds directly, including in the control model effects caused by rotor lift and drag coefficients.

If one neglect rotor accelerations and define the virtual torque executed by the rotors as

$$\boldsymbol{\Gamma}_B = b(t) + \sum_1^N I_i \dot{\omega}_i \mathbf{n}_i, \quad (3.41)$$

it is possible to obtain a linear relationship, like in Equation (3.8), between the virtual torque and rotor speeds. Making

$$M_t = \begin{bmatrix} (C_{L1}(\mathbf{r}_1 \times \mathbf{n}_1) - C_{D1}d_1 \mathbf{n}_1)^T \\ \vdots \\ (C_{LN}(\mathbf{r}_N \times \mathbf{n}_N) - C_{DN}d_N \mathbf{n}_N)^T \end{bmatrix}^T, \quad (3.42)$$

where d_1, \dots, d_N indicates rotor right-hand rule rotation directions in relation to $\mathbf{n}_1, \dots, \mathbf{n}_N$, one obtains

$$\boldsymbol{\Gamma}_B^d = M_t \boldsymbol{\Omega}. \quad (3.43)$$

The "Direct" modification is obtained, therefore, by making

$$\begin{aligned} f(x) &\leftarrow \begin{bmatrix} I_{cg}^{-1} [(I_{cg}\boldsymbol{\omega}_B - a(t)) \times \boldsymbol{\omega}_B] \\ Q^{-1}\mathbf{g}_A - \mu_B Q^{-1} \frac{\mathbf{v}_A}{m} \end{bmatrix} & g(x) &\leftarrow \begin{bmatrix} I_{cg}^{-1} M_t \\ M_f/m \end{bmatrix} & \ddot{x} &\leftarrow \begin{bmatrix} \dot{\boldsymbol{\omega}}_B \\ \dot{\mathbf{v}}_B \end{bmatrix} \\ \ddot{x}_d &\leftarrow \begin{bmatrix} \dot{\boldsymbol{\omega}}_B^d \\ Q^{-1} \frac{\mathbf{T}_A^d}{m} \end{bmatrix} & \dot{e} &\leftarrow \begin{bmatrix} \boldsymbol{\omega}_B^d - \boldsymbol{\omega}_B \\ Q^{-1} \frac{\mathbf{T}_A^d}{m} \delta t \end{bmatrix} & e &\leftarrow \begin{bmatrix} \mathbf{q}_e \\ \mathbf{e}_{P_B} \end{bmatrix} & u \rightarrow \boldsymbol{\Omega}^d \end{aligned} \quad (3.44)$$

Equation (3.26) requires the inverse of $g(x)$ to calculate the control input in the control surface. However, considering that $g(x)$ is not square in this case, the inverse is not easily obtained. Equation (3.26) can, therefore, be substituted by the same transformation used in Equation (3.9), but operating over the vector $\ddot{x}_d + c\dot{e} - f(x)$.

The transformation which we chose here is the control allocation method of Section 3.5.3. It is interesting to note that even though the SMC-D generates rotor speeds as its outputs, it is still necessary to use a CA method for inverting $g(x)$, even though the output of this inversion will not result in rotor speeds like in a conventional control allocation.

Finally, SMC-D results in the rotor speeds to be passed to actuator-level control after recasting rotation directions by Equation (3.121).

3.4.2.6 SOSMC Active Direct (ASMC-D)

In the SMC-D modification, rotor parameters are included not only in $a(t)$ but also in matrices M_f and M_t . Let

$$\Lambda_{fdd} = \text{diag}(\lambda_{i,fdd}), \text{ for } i = 1, \dots, N \quad (3.45)$$

be the matrix representation of rotor failure detection and diagnosis with $\lambda_{i,fdd}$ as proposed in Section 4.1.6 similar to Section 3.4.2.2. Online controller updating by a FDD algorithm can be implemented in the SMC-D by making

$$\begin{aligned} f(x) &\leftarrow \begin{bmatrix} I_{cg}^{-1} [(I_{cg}\boldsymbol{\omega}_B - a_{fdd}(t)) \times \boldsymbol{\omega}_B] \\ Q^{-1}\mathbf{g}_A - \mu_B Q^{-1}\frac{\mathbf{v}_A}{m} \end{bmatrix} & g(x) &\leftarrow \begin{bmatrix} I_{cg}^{-1} M_t \Lambda_{fdd} \\ M_f \Lambda_{fdd}/m \end{bmatrix} & \ddot{x} &\leftarrow \begin{bmatrix} \dot{\boldsymbol{\omega}}_B \\ \dot{\mathbf{v}}_B \end{bmatrix} \\ \ddot{x}_d &\leftarrow \begin{bmatrix} \dot{\boldsymbol{\omega}}_B^d \\ Q^{-1}\frac{\mathbf{T}_A^d}{m} \end{bmatrix} & \dot{e} &\leftarrow \begin{bmatrix} \boldsymbol{\omega}_B^d - \boldsymbol{\omega}_B \\ Q^{-1}\frac{\mathbf{T}_A^d}{m}\delta t \end{bmatrix} & e &\leftarrow \begin{bmatrix} \mathbf{q}_e \\ \mathbf{e}_{P_B} \end{bmatrix} & u \rightarrow \boldsymbol{\Omega}^d \end{aligned} \quad (3.46)$$

The same procedure as in Section 3.4.2.5 applies here to obtain desired rotor speeds.

3.4.3 Adaptive

According to notation used in (SCHWAGER; ANNASWAMY; LAVRETSKY, 2005), let

$$\dot{x} = A_p x + B_p \Lambda u + B_p f, \quad (3.47)$$

be the state-space description of a system to be controlled where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the system input, $A_p \in \mathbb{R}^{n \times n}$ is unknown, $B_p \in \mathbb{R}^{n \times m}$ is known and input failures are represented by $\Lambda \in \mathbb{R}^{m \times m}$ and $f \in \mathbb{R}^m$. Λ is a diagonal matrix with elements $0 \leq \lambda_{ii} \leq 1$, for $i = 1, \dots, m$, where $\lambda_{ii} = 0$ represents a total failure of input i while $\lambda_{ii} = 1$ represents its normal behaviour. Vector f represents either a failure where one input keeps its state unchanged or gets disturbed.

Let

$$\dot{x}_m = A_m x_m + B_m r \quad (3.48)$$

be a reference model which one want states of Equation (3.47) to follow to based on a control input u . $x_m \in \mathbb{R}^n$ is the reference state, $r \in \mathbb{R}^l$ is a pilot command with $l \leq m$, $A_m \in \mathbb{R}^{n \times n}$ is Hurwitz and $B_m \in \mathbb{R}^{n \times l}$. Let also u_{max_i} and u_{min_i} be the maximum and minimum control input commands for input i . Assume a control law

$$\begin{aligned} u &= \text{sat}(u_c), \\ u_c &= \hat{K}_x x + \hat{K}_r r + \hat{f} \end{aligned} \quad (3.49)$$

such that

$$\text{sat}_i(u_{ci}) = \begin{cases} u_{min_i}, & \text{se } u_i < u_{min_i} \\ u_i, & \text{se } u_{min_i} \leq u_i \leq u_{max_i} \\ u_{max_i}, & \text{se } u_i > u_{max_i}, \text{ para } i = 1, \dots, m \end{cases} \quad (3.50)$$

is a control input saturation, $\hat{K}_x \in \mathbb{R}^{m \times n}$, $\hat{K}_r \in \mathbb{R}^{m \times l}$ and $\hat{f} \in \mathbb{R}^m$. Substituting Equation (3.49) in Equation (3.47), one gets

$$\dot{x} = A_p x + B_p \Lambda [\hat{K}_x x + \hat{K}_r r + \hat{f}] + B_p \Lambda \Delta u + B_p f, \quad (3.51)$$

where

$$\Delta u = u - u_c \quad (3.52)$$

is the actuator saturation error.

For a perfect model following, Equation (3.53) can be assumed.

$$\begin{aligned} A_p + B_p \Lambda K_x^* &= A_m \\ B_p \Lambda K_r^* &= B_m \\ B_p (\Lambda f^* + f) &= 0, \end{aligned} \quad (3.53)$$

where K_x^* , K_r^* and Λf^* are optimal gain values to guarantee perfect model following.

Define the estimation errors $\tilde{K}_x = \hat{K}_x - K_x^*$, $\tilde{K}_r = \hat{K}_r - K_r^*$ and $\tilde{f} = \hat{f} - f^*$ and the model following error

$$e = x - x_m. \quad (3.54)$$

One can describe the model following error dynamics by

$$\dot{e} = A_m e + B_p \Lambda [\tilde{K}_x + \tilde{K}_r + \tilde{f} + \Delta u]. \quad (3.55)$$

Let

$$\dot{e}_\Lambda \doteq A_m e_\Lambda + B_p \text{diag}(\hat{\lambda}) \Delta u \quad (3.56)$$

be the error caused by input saturation where $\text{diag}(\hat{\lambda}) = \hat{\Lambda}$. The dynamics of the estimation error

$$e_u = e - e_\Lambda \quad (3.57)$$

can be obtained by

$$\dot{e}_u = A_m e_u + B_p \Lambda (\tilde{K}_x x + \tilde{K}_r r + \tilde{f}) + B_p \text{diag}(\Delta u) \tilde{\lambda}, \quad (3.58)$$

where $\tilde{\lambda} = \lambda - \hat{\lambda}$.

Define the Lyapunov function

$$V \doteq e_u^T P e_u + \text{tr}(\tilde{K}_x^T \Gamma_1^{-1} \Lambda \tilde{K}_x) + \text{tr}(\tilde{K}_r^T \Gamma_2^{-1} \Lambda \tilde{K}_r) + \tilde{f}^T \Gamma_3^{-1} \Lambda \tilde{f} + \tilde{\lambda}^T \Gamma_4^{-1} \tilde{\lambda}, \quad (3.59)$$

with first-order derivative

$$\begin{aligned} \dot{V} &= e_u^T [A_m^T P + P A_m] e_u + \\ &\quad + 2 \text{tr}[\tilde{K}_x^T \Lambda (B_p^T P e_u x^T + \Gamma_1^{-1} \dot{\tilde{K}}_x)] + \\ &\quad + 2 \text{tr}[\tilde{K}_r^T \Lambda (B_p^T P e_u r^T + \Gamma_2^{-1} \dot{\tilde{K}}_r)] + \\ &\quad + 2 \tilde{f}^T \Lambda (B_p^T P e_u + \Gamma_3^{-1} \dot{\tilde{f}}) + \\ &\quad + 2 \tilde{\lambda}^T \Lambda (\text{diag}(\Delta u) B_p^T P e_u + \Gamma_4^{-1} \dot{\tilde{\lambda}}). \end{aligned} \quad (3.60)$$

The error convergence is guaranteed for $\dot{V} < 0$ (IOANNOU; SUN, 2012), which can be obtained by making

$$A_m^T P + PA_m = -Q, \quad (3.61)$$

$$\dot{\hat{K}}_x = -\Gamma_1 B_P^T P e_u x^T, \quad (3.62)$$

$$\dot{\hat{K}}_r = -\Gamma_2 B_P^T P e_u r^T, \quad (3.63)$$

$$\dot{\hat{f}} = -\Gamma_3 B_P^T P e_u, \text{ and} \quad (3.64)$$

$$\dot{\hat{\lambda}}_x = -\Gamma_4 \text{diag}(\Delta u) B_p^T P e_u, \quad (3.65)$$

where Γ_i for $i = 1, \dots, 4$ and Q are definite positive matrices. Considering that K_x^* , K_r^* , f^* and λ^* are constant, their derivatives are null. From the estimation errors definition, one get $\dot{\hat{K}}_x = \dot{K}_x$, $\dot{\hat{K}}_r = \dot{K}_r$, $\dot{\hat{f}} = \dot{f}$ and $\dot{\hat{\lambda}} = -\dot{\hat{\lambda}}$, which gives the adaptation law

$$\dot{\hat{K}}_x = -\Gamma_1 B_P^T P e_u x^T, \quad (3.66)$$

$$\dot{\hat{K}}_r = -\Gamma_2 B_P^T P e_u r^T, \quad (3.67)$$

$$\dot{\hat{f}} = -\Gamma_3 B_P^T P e_u, \text{ and} \quad (3.68)$$

$$\dot{\hat{\lambda}}_x = \Gamma_4 \text{diag}(\Delta u) B_p^T P e_u. \quad (3.69)$$

3.4.3.1 Adaptive virtual torque control (MRAC)

Using the modelling notation of this work, one can apply the Adaptive controller for multirotor attitude control by making

$$\begin{array}{llll} x \leftarrow \omega_B & x_m \leftarrow \omega_m & r \leftarrow \dot{\omega}_B^d & B_m \leftarrow 1_{3 \times 3} \\ A_m \leftarrow A_m & B_p \leftarrow I_{cg}^{-1} & e \leftarrow \omega_B - \omega_m & u \rightarrow \Gamma_B^d \end{array} \quad (3.70)$$

3.4.3.2 Adaptive control with PID following (MRAC-P)

The modification presented here is similar to the one in Section 3.4.2.3 but applied to the Adaptive controller.

The reference model must be adapted to consider the gravitational acceleration, hence following the structure

$$\dot{x}_m = A_m x_m + B_m r + C, \quad (3.71)$$

for $C \in \mathbb{R}^n$. To obtain x_m , one can solve

$$\dot{z}_m = A_m z_m + B_m r \quad (3.72)$$

and make

$$x_m = z_m - A_m^{-1} C. \quad (3.73)$$

Using the same notation as in Section 3.4.3.1, the MRAC-P can be applied to multirotor control by making

$$\begin{aligned} x &\leftarrow \begin{bmatrix} \mathbf{v}_B \\ \boldsymbol{\omega}_B \end{bmatrix} & x_m &\leftarrow \begin{bmatrix} \mathbf{v}_m \\ \boldsymbol{\omega}_m \end{bmatrix} & r &\leftarrow \begin{bmatrix} Q^{-1}\mathbf{T}_A^d \\ \boldsymbol{\omega}_B^d \end{bmatrix} & B_m &\leftarrow \begin{bmatrix} 1_{3x3}/m & 0_{3x3} \\ 0_{3x3} & 1_{3x3} \end{bmatrix} & C &\leftarrow \begin{bmatrix} \mathbf{g}_B \\ 0_{3x1} \end{bmatrix} \\ A_m &\leftarrow A_m & B_p &\leftarrow \begin{bmatrix} 1_{3x3}/m & 0_{3x3} \\ 0_{3x3} & I_{cg}^{-1} \end{bmatrix} & e &\leftarrow \begin{bmatrix} \mathbf{v}_B - \mathbf{v}_m \\ \boldsymbol{\omega}_B - \boldsymbol{\omega}_m \end{bmatrix} & u &\rightarrow \begin{bmatrix} Q^{-1}\mathbf{T}_A^{d*} \\ \boldsymbol{\Gamma}_B^d \end{bmatrix}, \end{aligned} \quad (3.74)$$

where \mathbf{T}_A^{d*} shall substitute \mathbf{T}_A^d in next steps of control allocation.

3.4.3.3 Adaptive Direct control (MRAC-D)

The same idea of Section 3.4.2.5 is considered here but applied to the Adaptive controller.

Using the same notation and similar procedures as in Section 3.4.3.2, the MRAC-D can be applied to multirotor control by making

$$\begin{aligned} x &\leftarrow \begin{bmatrix} \mathbf{v}_B \\ \boldsymbol{\omega}_B \end{bmatrix} & x_m &\leftarrow \begin{bmatrix} \mathbf{v}_m \\ \boldsymbol{\omega}_m \end{bmatrix} & r &\leftarrow \begin{bmatrix} Q^{-1}\mathbf{T}_A^d \\ \boldsymbol{\omega}_B^d \end{bmatrix} & B_m &\leftarrow \begin{bmatrix} 1_{3x3}/m & 0_{3x3} \\ 0_{3x3} & 1_{3x3} \end{bmatrix} & C &\leftarrow \begin{bmatrix} \mathbf{g}_B \\ 0_{3x1} \end{bmatrix} \\ A_m &\leftarrow A_m & B_p &\leftarrow \begin{bmatrix} M_f/m \\ I_{cg}^{-1}M_t \end{bmatrix} & e &\leftarrow \begin{bmatrix} \mathbf{v}_B - \mathbf{v}_m \\ \boldsymbol{\omega}_B - \boldsymbol{\omega}_m \end{bmatrix} & u &\rightarrow \boldsymbol{\Omega}^d. \end{aligned} \quad (3.75)$$

Finally, Adaptive Direct controller results in the rotor speeds to be passed to actuator-level control after recasting rotation directions by Equation (3.121).

3.4.4 R-LQR

Following notation of (CERRI, 2009), let

$$x_{i+1} = (F_i + \delta F_i)x_i + (G_i + \delta G_i)u_i, \quad i = 0, \dots, N, \quad (3.76)$$

be a linear discrete-time system subject to parametric uncertainties where $F_i \in \mathbb{R}^{n \times n}$ and $G_i \in \mathbb{R}^{n \times m}$ are discrete state-space matrices, $x_i \in \mathbb{R}^n$ is the state vector, $u_i \in \mathbb{R}^m$ is the control input and x_0 is the known initial state. Let $\delta F_i \in \mathbb{R}^{n \times n}$ and $\delta G_i \in \mathbb{R}^{n \times m}$ be uncertainties matrices modelled by

$$[\delta F_i \quad \delta G_i] = H_i \Delta_i [E_{F_i} \quad E_{G_i}]; \quad i = 0, \dots, N, \quad (3.77)$$

with $H_i \in \mathbb{R}^{n \times p}$, $E_{F_i} \in \mathbb{R}^{l \times n}$, $E_{G_i} \in \mathbb{R}^{l \times m}$ known matrices and $\Delta_i \in \mathbb{R}^{p \times l}$ a contraction matrix with $\|\Delta_i\| \leq 1$. Consider H_i not zero for all i .

The Robust Linear Quadratic Regulator, as introduced by (CERRI, 2009), is summarized on Table 1.

In multirotor control there is no previous knowledge of matrices in Equations (3.76) and (3.77), unless a linearization is made over an operational point. This happens

Table 1: Robust Linear Quadratic Regulator

Uncertain model: Consider Equations (3.76) and (3.77) with $F_i, G_i, E_{F_i}, E_{G_i}$, $Q_i \succ 0$ and $R_i \succ 0$ known for all i .

Initial conditions: Define x_0 and $P_{N+1} \succ 0$.

Step 1: (Backwards)

Calculate for all $i = N, \dots, 0$:

$$\begin{bmatrix} L_i \\ K_i \\ P_i \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -I \\ 0 & 0 & F_i \\ 0 & 0 & E_{F_i} \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}^T \begin{bmatrix} P_{i+1}^{-1} & 0 & 0 & 0 & 0 & I & 0 \\ 0 & R_i^{-1} & 0 & 0 & 0 & 0 & I \\ 0 & 0 & Q_i^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & -G_i \\ 0 & 0 & 0 & 0 & 0 & 0 & -E_{G_i} \\ I & 0 & 0 & I & 0 & 0 & 0 \\ 0 & I & 0 & -G_i^T & -E_{G_i}^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ -I \\ F_i \\ E_{F_i} \\ 0 \\ 0 \end{bmatrix}.$$

Step 2: (Forwards)

Calculate for all $i = 0, \dots, N$:

$$\begin{bmatrix} x_{i+1}^* \\ u_i^* \end{bmatrix} = \begin{bmatrix} L_i \\ K_i \end{bmatrix} x_i^*,$$

with total cost given by $J_r^* = x_0^T P_0 x_0$.

because the system matrices assume different values depending on the multirotor state, as will be shown below. Therefore, it is not possible to control the system following the order presented on Table 1. In this work, R-LQR calculation is made always forwards as presented on Table 2 and assuming the Riccati P_i will stabilize.

3.4.4.1 R-LQR virtual torque control (R-LQR)

The regulator presented on Section 3.4.4 needs to be adapted so as to follow a reference for virtual torque control. Let

$$A(\boldsymbol{\omega}_B, \omega_i) = I_{cg}^{-1}[I_{cg}\boldsymbol{\omega}_B - a(t)]_\times, \quad (3.78)$$

where $[.]_\times$ represents the Skew-Symmetric matrix transformation which turns Equation (2.24) into

$$\dot{\boldsymbol{\omega}}_B = A(\boldsymbol{\omega}_B, \omega_i)\boldsymbol{\omega}_B + I_{cg}^{-1}b(t). \quad (3.79)$$

$b(t)$ here is the same as $\boldsymbol{\Gamma}_B$ by definition. Despite the adaptation for quaternions included in this work, (LEAO, 2015) defined a reference following error to apply the R-LQR for

Table 2: Robust Linear Quadratic Regulator - Always-forwards calculation

Uncertain model: Consider Equations (3.76) and (3.77) with $F_i, G_i, E_{F_i}, E_{G_i}$, $Q_i \succ 0$ and $R_i \succ 0$ known for all i .

Initial conditions: Define x_0 and $P_1 \succ 0$.

Step 1:

Calculate for $i = 1$:

$$\begin{bmatrix} L_i \\ K_i \\ P_i \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -I \\ 0 & 0 & F_i \\ 0 & 0 & E_{F_i} \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}^T \begin{bmatrix} P_{i-1}^{-1} & 0 & 0 & 0 & 0 & I & 0 \\ 0 & R_i^{-1} & 0 & 0 & 0 & 0 & I \\ 0 & 0 & Q_i^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & -G_i \\ 0 & 0 & 0 & 0 & 0 & 0 & -E_{G_i} \\ I & 0 & 0 & I & 0 & 0 & 0 \\ 0 & I & 0 & -G_i^T & -E_{G_i}^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ -I \\ F_i \\ E_{F_i} \\ 0 \\ 0 \end{bmatrix}.$$

Step 2:

Calculate for $i = 1$:

$$\begin{bmatrix} x_{i+1}^* \\ u_i^* \end{bmatrix} = \begin{bmatrix} L_i \\ K_i \end{bmatrix} x_i,$$

Repeat Steps 1 and 2 for all $i = 2, \dots, N$.

multirotor attitude control by making

$$-\dot{\omega}_B + \dot{\omega}_B^d = -A(\omega_B, \omega_i)\omega_B + A(\omega_B, \omega_i)\omega_B^d - I_{cg}^{-1}\Gamma_B + \dot{\omega}_B^d - A(\omega_B, \omega_i)\omega_B^d \quad (3.80)$$

$$u_i^* \doteq -I_{cg}^{-1}\Gamma_B + \dot{\omega}_B^d - A(\omega_B, \omega_i)\omega_B^d \quad (3.81)$$

which results in

$$\begin{aligned} x_i &\leftarrow \begin{bmatrix} \omega_B^d - \omega_B \\ q_e \end{bmatrix} & F_i &\leftarrow c2d \left(\begin{bmatrix} A(\omega_B, \omega_i) & 0_{3x3} \\ \frac{1}{2}S(q)_{(2:4,:)} & 0_{3x3} \end{bmatrix}, \delta t \right) \\ G_i &\leftarrow c2d \left(\begin{bmatrix} 1_{3x3} \\ 0_{3x3} \end{bmatrix}, \delta t \right) & -I_{cg}(u_i^* - \dot{\omega}_B^d + A(\omega_B, \omega_i)\omega_B^d) &\rightarrow \Gamma_B^d, \end{aligned} \quad (3.82)$$

where $c2d(\bullet, \delta t)$ represents a discretizing function over control loop time δt for the system $F_i, G_i, S(q)_{(2:4,:)}$ represents the three last lines of $S(q)$ and all other variables follow the notation already used in this work.

3.4.4.2 R-LQR Active virtual torque control (AR-LQR)

Consider the same FDD online updating as proposed in Section 3.4.2.2. Equation (3.78) becomes

$$A_{fdd}(\omega_B, \omega_i) = I_{cg}^{-1}[I_{cg}\omega_B - a_{fdd}(t)]\times, \quad (3.83)$$

which yields the following adaptation for a R-LQR Active virtual torque control for multirotors

$$\begin{aligned} x_i &\leftarrow \begin{bmatrix} \boldsymbol{\omega}_B^d - \boldsymbol{\omega}_B \\ \mathbf{q}_e \end{bmatrix} & F_i &\leftarrow c2d \left(\begin{bmatrix} A_{fdd}(\boldsymbol{\omega}_B, \omega_i) & 0_{3x3} \\ \frac{1}{2}S(q)_{(2:4,:)} & 0_{3x3} \end{bmatrix}, \delta t \right) \\ G_i &\leftarrow c2d \left(\begin{bmatrix} 1_{3x3} \\ 0_{3x3} \end{bmatrix}, \delta t \right) & -I_{cg} (u_i^* - \dot{\boldsymbol{\omega}}_B^d + A_{fdd}(\boldsymbol{\omega}_B, \omega_i)\boldsymbol{\omega}_B^d) &\rightarrow \boldsymbol{\Gamma}_B^d. \end{aligned} \quad (3.84)$$

3.4.4.3 R-LQR virtual torque control with rotor failures (FT-LQR)

Equation (3.82) makes clear that there is no information about the rotors in the proposed controller, which do not allow the algorithm to consider rotor parameter variations. The proposed here is to make

$$\dot{\boldsymbol{\omega}}_B = A(\boldsymbol{\omega}_B, \omega_i)\boldsymbol{\omega}_B + I_{cg}^{-1}M_t\boldsymbol{\Omega} \quad (3.85)$$

$$\dot{\boldsymbol{\omega}}_B + \dot{\boldsymbol{\omega}}_B^d = -A(\boldsymbol{\omega}_B, \omega_i)\boldsymbol{\omega}_B + A(\boldsymbol{\omega}_B, \omega_i)\boldsymbol{\omega}_B^d - I_{cg}^{-1}M_t\boldsymbol{\Omega} + \dot{\boldsymbol{\omega}}_B^d - A(\boldsymbol{\omega}_B, \omega_i)\boldsymbol{\omega}_B^d \quad (3.86)$$

$$-I_{cg}^{-1}M_t u_i^* \dot{=} -I_{cg}^{-1}M_t\boldsymbol{\Omega} + \dot{\boldsymbol{\omega}}_B^d - A(\boldsymbol{\omega}_B, \omega_i)\boldsymbol{\omega}_B^d, \quad (3.87)$$

resulting in

$$\begin{aligned} x_i &\leftarrow \begin{bmatrix} \boldsymbol{\omega}_B^d - \boldsymbol{\omega}_B \\ \mathbf{q}_e \end{bmatrix} & F_i &\leftarrow c2d \left(\begin{bmatrix} A(\boldsymbol{\omega}_B, \omega_i) & 0_{3x3} \\ \frac{1}{2}S(q)_{(2:4,:)} & 0_{3x3} \end{bmatrix}, \delta t \right) \\ G_i &\leftarrow c2d \left(\begin{bmatrix} -I_{cg}^{-1}M_t \\ 0_{3x3} \end{bmatrix}, \delta t \right) & -I_{cg}^{-1} (-I_{cg}^{-1}M_t u_i^* - \dot{\boldsymbol{\omega}}_B^d + A(\boldsymbol{\omega}_B, \omega_i)\boldsymbol{\omega}_B^d) &\rightarrow \boldsymbol{\Gamma}_B^d. \end{aligned} \quad (3.88)$$

Since M_t is present in G_i , rotor parameter variations can be included in E_{G_i} . This is expected to improve controller robustness to rotor failures even though these failures will not be included in the calculation of $\boldsymbol{\Gamma}_B^d$.

3.4.4.4 R-LQR Active virtual torque control with rotor failures (AFT-LQR)

Following the same representation of rotor failure detection and diagnosis as in Section 3.4.2.6, online controller updating for the presented on Section 3.4.4.3 can be obtained by making

$$\begin{aligned} F_i &\leftarrow c2d \left(\begin{bmatrix} A_{fdd}(\boldsymbol{\omega}_B, \omega_i) & 0_{3x3} \\ \frac{1}{2}S(q)_{(2:4,:)} & 0_{3x3} \end{bmatrix}, \delta t \right) & G_i &\leftarrow c2d \left(\begin{bmatrix} -I_{cg}^{-1}M_t\Lambda_{fdd} \\ 0_{3x3} \end{bmatrix}, \delta t \right) \quad (3.89) \\ x_i &\leftarrow \begin{bmatrix} \boldsymbol{\omega}_B^d - \boldsymbol{\omega}_B \\ \mathbf{q}_e \end{bmatrix} & -I_{cg}^{-1} (-I_{cg}^{-1}M_t\Lambda_{fdd} u_i^* - \dot{\boldsymbol{\omega}}_B^d + A_{fdd}(\boldsymbol{\omega}_B, \omega_i)\boldsymbol{\omega}_B^d) &\rightarrow \boldsymbol{\Gamma}_B^d. \end{aligned}$$

3.4.4.5 R-LQR with rotor failures and PIDD following (FT-LQR-P)

Using the same proposition as in Sections 3.4.2.3 and 3.4.3.2 the R-LQR with PIDD can be applied to multirotor control by making

$$\begin{aligned} x_i &\leftarrow \begin{bmatrix} \mathbf{v}_B^d - \mathbf{v}_B \\ \boldsymbol{\omega}_B^d - \boldsymbol{\omega}_B \\ \mathbf{q}_e \end{bmatrix} \quad G_i \leftarrow c2d \left(\begin{bmatrix} M_f/m \\ -I_{cg}^{-1}M_t \\ 0_{3x3} \end{bmatrix}, \delta t \right) \\ F_i &\leftarrow c2d \left(\begin{bmatrix} -\mu_B/m & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & A(\boldsymbol{\omega}_B, \omega_i) & 0_{3x3} \\ 0_{3x3} & \frac{1}{2}S(q)_{(2:4,:)} & 0_{3x3} \end{bmatrix}, \delta t \right) \\ \begin{bmatrix} -I_{cg}^{-1}(-I_{cg}^{-1}M_t u_i^* - \dot{\boldsymbol{\omega}}_B^d + A(\boldsymbol{\omega}_B, \omega_i)\boldsymbol{\omega}_B^d) \\ m\dot{\mathbf{v}}_B^d - M_f u_i^* + \mu_B \mathbf{v}_B^d + mQ^{-1}\mathbf{g}_A \end{bmatrix} &\rightarrow \begin{bmatrix} \boldsymbol{\Gamma}_B^d \\ Q^{-1}\mathbf{T}_A^{d*} \end{bmatrix}, \end{aligned} \quad (3.90)$$

where \mathbf{T}_A^{d*} shall substitute \mathbf{T}_A^d in next steps of control allocation. In this case, both M_f and M_t are present in the controller and rotor parameter variations can be included in E_{G_i} while linear parameter variations can be included in E_{F_i} .

3.4.4.6 R-LQR Active with rotor failures and PIDD following (AFT-LQR-P)

FDD can be included in the controller of Section 3.4.4.5 by making

$$\begin{aligned} x_i &\leftarrow \begin{bmatrix} \mathbf{v}_B^d - \mathbf{v}_B \\ \boldsymbol{\omega}_B^d - \boldsymbol{\omega}_B \\ \mathbf{q}_e \end{bmatrix} \quad G_i \leftarrow c2d \left(\begin{bmatrix} M_f\Lambda_{fdd}/m \\ -I_{cg}^{-1}M_t\Lambda_{fdd} \\ 0_{3x3} \end{bmatrix}, \delta t \right) \\ F_i &\leftarrow c2d \left(\begin{bmatrix} -\mu_B/m & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & A_{fdd}(\boldsymbol{\omega}_B, \omega_i) & 0_{3x3} \\ 0_{3x3} & \frac{1}{2}S(q)_{(2:4,:)} & 0_{3x3} \end{bmatrix}, \delta t \right) \\ \begin{bmatrix} -I_{cg}^{-1}(-I_{cg}^{-1}M_t\Lambda_{fdd} u_i^* - \dot{\boldsymbol{\omega}}_B^d + A_{fdd}(\boldsymbol{\omega}_B, \omega_i)\boldsymbol{\omega}_B^d) \\ m\dot{\mathbf{v}}_B^d - M_f\Lambda_{fdd} u_i^* + \mu_B \mathbf{v}_B^d + mQ^{-1}\mathbf{g}_A \end{bmatrix} &\rightarrow \begin{bmatrix} \boldsymbol{\Gamma}_B^d \\ Q^{-1}\mathbf{T}_A^{d*} \end{bmatrix}, \end{aligned} \quad (3.91)$$

where \mathbf{T}_A^{d*} shall substitute \mathbf{T}_A^d in next steps of control allocation.

3.4.5 Robust Regulator for Discrete-time Markovian Jump Linear Systems

This section will present two approaches from the view of Markovian systems, as cited on Section 1.2. Section 3.4.5.1 presents the application of a mode-dependent regulator, or according to the nomenclature used here, an "active" regulator, for multirotor control under rotor failures and considering uncertain probability transition matrix, which makes it a "passive" controller. Section 3.4.6 presents the same application but for a regulator with uncertain or unknown transition probability matrix and unknown mode via Singular Augmented System. In both cases each rotor failure characterizes a different Markovian mode.

3.4.5.1 Mode-dependent (AM-LQR)

The Robust Linear-Quadratic regulator for discrete-time Markovian jump linear systems subject to uncertainties and uncertain probability transition matrix (MLSU) can be obtained by the presented below considering notation as presented by (BORTOLIN, 2017).

Consider the following MLSU

$$x_{k+1} = (F_{\theta(k)} + \delta F_{\theta(k)})x_k + (G_{\theta(k)} + \delta G_{\theta(k)})u_k, \quad \forall k \geq 0, \quad \theta(0) \sim \pi_0 \quad (3.92)$$

where $(x_k, \theta(k))$ is the known system state and $u_k \in \mathbb{R}^m$ is the control input. $\theta = \{\theta_{(0)}, \dots, \theta_{(N-1)}\}$ is a Markov chain with $\theta_{(k)} \in \mathcal{S} := \{1, \dots, s\}$ related to a transition probability matrix $\mathbb{P}^\delta := [\mathbb{P} + \delta \mathbb{P}] = [p_{ij} + \delta p_{ij}] \in \mathbb{R}^{s \times s}$ defined by

$$\begin{aligned} p_{ij}^\delta &= \text{Prob} \left[\theta_{(k+1)} = j \mid \theta_{(k)} = i \right] = p_{ij} + \delta p_{ij}, \\ \sum_{j=1}^s (p_{ij} + \delta p_{ij}) &= 1 \quad \text{and} \quad 0 \leq p_{ij} + \delta p_{ij} \leq 1, \end{aligned} \quad (3.93)$$

where \mathbb{P} is a nominal transition probability matrix and $\delta \mathbb{P}$ an uncertainty matrix. The uncertain probabilities can be modelled by

$$\delta p_{ij} = h_{ij} \Delta_{ij} e_{ij}, \quad (3.94)$$

where h_{ij} and e_{ij} are known scalars and Δ_{ij} is a scalar in the interval $[\underline{\gamma}_{ij}, \bar{\gamma}_{ij}]$.

In (3.92), $F_{\theta(k)} \in \mathbb{R}^{n \times n}$ and $G_{\theta(k)} \in \mathbb{R}^{n \times m}$ are known nominal system matrices; and $\delta F_{\theta(k)} \in \mathbb{R}^{n \times n}$ and $\delta G_{\theta(k)} \in \mathbb{R}^{n \times m}$ are uncertainty matrices modelled by

$$\begin{bmatrix} \delta F_{\theta(k)} & \delta G_{\theta(k)} \end{bmatrix} = H_{\theta(k)} \Delta_{\theta(k)} \begin{bmatrix} E_{F_{\theta(k)}} & E_{G_{\theta(k)}} \end{bmatrix}, \quad \forall k \geq 0, \quad (3.95)$$

with $H_{\theta(k)} \in \mathbb{R}^{n \times t}$ (non null matrix); $E_{F_{\theta(k)}} \in \mathbb{R}^{l \times n}$ and $E_{G_{\theta(k)}} \in \mathbb{R}^{l \times m}$ known; and $\Delta_{\theta(k)} \in \mathbb{R}^{t \times l}$ such that $\|\Delta_{\theta(k)}\| \leq 1$, for all $i \in \mathcal{S}$.

The purpose is to design an optimal feedback control sequence $\{u_k^*\}_{k=0}^{N-1}$ that regulates the MLSU (3.92), based on a control law $u_k = K_{\theta(k),k} x_k$, for $K_{\theta(k),k} \in \mathbb{R}^{m \times n}$, with uncertainties defined in Equations (3.94) and (3.104).

One can define the optimization problem for this regulator class in the following manner:

$$\min_{x_{k+1}, u_k} \max_{\delta F_{\theta(k)}, \delta G_{\theta(k)}} \mathfrak{J}_{\mu,k}(\theta(k), x_{k+1}, u_k), \quad (3.96)$$

for all $k = N - 1, \dots, 0$ and $\theta_{(k)} = i \in \mathcal{S}$, where the penalized regularized quadratic cost function $\mathfrak{J}_{\mu,k}(.)$ is defined by (3.97) with $\mathcal{F}_{\theta(k)} := (F_{\theta(k)} + \delta F_{\theta(k)})$, $\mathcal{G}_{\theta(k)} := (G_{\theta(k)} + \delta G_{\theta(k)})$,

$$\Psi_{\theta(k),k+1} = \sum_{j=1}^s P_{j,k+1} p_{\theta(k)j}, \mathcal{P}_{k+1} = \text{diag}(P_{i,k+1}), R_{\theta(k)} \succ 0 \text{ and } Q_{\theta(k)} \succ 0.$$

$$\begin{aligned} \mathfrak{J}_{\mu,k}(\theta(k), x_{k+1}, u_k) &= \begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix}^T \begin{bmatrix} \Psi_{\theta(k),k+1} & 0 \\ 0 & R_{\theta(k)} \end{bmatrix} \begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} + \\ &\quad \left(\begin{bmatrix} \delta_{\mathbf{p}_{\theta(k)}} & 0 \\ 0 & 0 \\ I_n & -\mathcal{G}_{\theta(k)}^\delta \end{bmatrix} \begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} - \begin{bmatrix} 0 \\ -I_n \\ \mathcal{F}_{\theta(k)}^\delta \end{bmatrix} x_k \right)^T \begin{bmatrix} \mathcal{P}_{k+1} & 0 & 0 \\ 0 & Q_{i,k} & 0 \\ 0 & 0 & \mu I_n \end{bmatrix} (\bullet) \end{aligned} \quad (3.97)$$

The uncertain matrix $\delta_{\mathbf{p}_{\theta(k)}}$ can be modelled by

$$\delta_{\mathbf{p}_i} = H_{\mathbf{p}_i} \Delta_{\mathbf{p}_i} E_{\mathbf{p}_i}, \quad (3.98)$$

where

$$H_{\mathbf{p}_i} = \begin{bmatrix} \sqrt{h_{i1}} I_n & 0 & \dots & 0 \\ 0 & \sqrt{h_{i2}} I_n & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{h_{is}} I_n \end{bmatrix}, \Delta_{\mathbf{p}_i} = \begin{bmatrix} \sqrt{\Delta_{i1}} I_n \\ \sqrt{\Delta_{i2}} I_n \\ \vdots \\ \sqrt{\Delta_{is}} I_n \end{bmatrix} \text{ and } E_{\mathbf{p}_i} = \sqrt{e_i} I_n \quad (3.99)$$

for all $\theta(k) = i$, where $H_{\mathbf{p}_i} \in \mathbb{R}^{sn \times sn}$ and $E_{\mathbf{p}_i} \in \mathbb{R}^{n \times n}$ are known matrices, $\Delta_{\mathbf{p}_i} \in \mathbb{R}^{sn \times n}$ is a contraction matrix such that $\|\Delta_{\mathbf{p}_i}\| \leq 1$ and the elements h_{ij} , Δ_{ij} and $e_i = e_{ij}$ are positive scalars. Matrices Q , R , and P are weight matrices and μ is a penalty factor that transforms the constrained minimization in an unconstrained one, guaranteeing that the equality (3.92) be true for all Markov states and guaranteeing the robust regularization of the LQR when the system is subject to Markovian jumps.

Omitting intermediary steps for brevity, the AM-LQR is presented on Table 3, where

$$\begin{aligned} \hat{I}_{\theta(k)} &= \begin{bmatrix} I_n \\ E_{\mathbf{p}_{\theta(k)}} \\ 0 \end{bmatrix}, \hat{F}_{\theta(k)} = \begin{bmatrix} F_{\theta(k)} \\ 0 \\ E_{F_{\theta(k)}} \end{bmatrix}, \hat{G}_{\theta(k)} = \begin{bmatrix} G_{\theta(k)} \\ 0 \\ E_{G_{\theta(k)}} \end{bmatrix}, \Sigma_{\theta(k)} = \begin{bmatrix} \Phi(\mu, \hat{\lambda}_\mu) & 0 & 0 \\ 0 & \hat{\lambda}_\mu^{-1} I_n & 0 \\ 0 & 0 & \hat{\lambda}_\mu^{-1} I_l \end{bmatrix} \\ \Phi(\mu, \hat{\lambda}_\mu) &= \mu^{-1} I_n - \hat{\lambda}_\mu^{-1} H_{\theta(k)} H_{\theta(k)}^T \text{ and } \hat{\lambda}_\mu = (1 + \alpha) \|\mu H_{\theta(k)}^T H_{\theta(k)}\|, \alpha > 0. \end{aligned} \quad (3.100)$$

When dealing with multirotors, the same case as in Section 3.4.4 applies for R-LQR for MLSU, which means it is not possible to generate control inputs from backwards as suggested on Step 1 of Table 3. Therefore, convergence of the probabilistic Riccati $\Psi_{\theta(k),k}$ is assumed when calculating control inputs as suggested in Table 4.

With the controller framework in hands, it is necessary to relate this framework to the model for multirotor control. In this case, the relation is similar to the one presented on Section 3.4.4.4. In Section 3.4.4.4, F_i and G_i were modified online depending on the FDD result. The difference here is that for each failure mode there is a different $\hat{F}_{\theta(k)}$ and

Table 3: Robust Regulator for MLSU

Initial conditions: Define $\mu > 0$, x_0 , π_0 , \mathcal{S} , N and $P_{\theta(N),N} \succ 0$.
Step 1: (<i>Backwards</i>) Calculate for all $k = N - 1, \dots, 0$:
$\Psi_{\theta(k),k+1} := \sum_{j=1}^s P_{j,k+1} p_{\theta(k)j} .$ $\begin{bmatrix} L_{\theta(k),k} \\ K_{\theta(k),k} \\ P_{\theta(k),k} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -I \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \hat{F}_{\theta(k)} \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}^T \begin{bmatrix} 0 & I & 0 & 0 & 0 & 0 & 0 & I & 0 \\ I & -\Psi_{\theta(k),k+1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & I \\ 0 & 0 & I & -R_{\theta(k)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & -Q_{\theta(k)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \Sigma_{\theta(k)} & \hat{I}_{\theta(k)} & -\hat{G}_{\theta(k)} \\ 0 & 0 & 0 & 0 & 0 & 0 & \hat{I}_{\theta(k)}^T & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & -\hat{G}_{\theta(k)}^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -I \\ 0 \\ 0 \\ \hat{F}_{\theta(k)} \\ 0 \\ 0 \end{bmatrix}$
Step 2: (<i>Forwards</i>) For each $k = 0, \dots, N - 1$, make:
$\begin{bmatrix} x_{k+1}^* \\ u_k^* \end{bmatrix} = \begin{bmatrix} L_{\theta(k),k} \\ K_{\theta(k),k} \end{bmatrix} x_k$

$\hat{G}_{\theta(k)}$, and all these matrices are used together to provide a probability estimated $\Psi_{\theta(k),k-1}$. The FDD is also used in this case to determine which markovian state $\theta(k)$ the system is at.

Let $\Lambda_{\theta(k)}$ and $A_{\theta(k)}$ be the same as in Equations (3.46) and (3.83), with the difference that $\theta(k)$ represent a failure mode from a set \mathcal{S} of previously defined failure modes. Hence, the multirotor control can be obtained by making

$$F_{\theta(k)} \leftarrow c2d \left(\begin{bmatrix} A_{\theta(k)}(\boldsymbol{\omega}_B, \omega_i) & 0_{3x3} \\ \frac{1}{2}S(q)_{(2:4,:)} & 0_{3x3} \end{bmatrix}, \delta t \right) \quad G_{\theta(k)} \leftarrow c2d \left(\begin{bmatrix} -I_{cg}^{-1} M_t \Lambda_{\theta(k)} \\ 0_{3x3} \end{bmatrix}, \delta t \right) \quad (3.101)$$

$$x_k \leftarrow \begin{bmatrix} \boldsymbol{\omega}_B^d - \boldsymbol{\omega}_B \\ \mathbf{q}_e \end{bmatrix} \quad -I_{cg}^{-1} (-I_{cg}^{-1} M_t \Lambda_j u_k^* - \dot{\boldsymbol{\omega}}_B^d + A_j(\boldsymbol{\omega}_B, \omega_i) \boldsymbol{\omega}_B^d) \rightarrow \boldsymbol{\Gamma}_B^d,$$

where $j \in \mathcal{S}$ is the failure mode identified by the FDD in the current control loop.

(BORTOLIN, 2017) remarks that the application of the above mentioned algorithm is conditioned to $\|\Delta\| \leq 1$, as presented above. When this condition is not satisfied, a scalar $\kappa \geq 1$ shall be considered, such that $\|\Delta\| \leq \kappa$ and following the equivalence

$$\Delta_{\mathbf{p}_{\theta(k)}} \leftarrow \frac{\Delta_{\mathbf{p}_{\theta(k)}}}{\kappa}, \quad \Delta_{\theta(k)} \leftarrow \frac{\Delta_{\theta(k)}}{\kappa},$$

$$E_{\mathbf{p}_{\theta(k)}} \leftarrow \kappa E_{\mathbf{p}_{\theta(k)}}, \quad E_{F\theta(k)} \leftarrow \kappa E_{F\theta(k)} \quad \text{and} \quad E_{G\theta(k)} \leftarrow \kappa E_{G\theta(k)}. \quad (3.102)$$

Table 4: Robust Regulator for MLSU considering only forwards calculation

Initial conditions: Define $\mu > 0$, x_0 , π_0 , \mathcal{S} , N and $P_{\theta(N),N} \succ 0$.

Step 1: Calculate for $k = 1$ and for $\theta(k) = 1, \dots, s \in \mathcal{S}$:

$$\begin{aligned} \Psi_{\theta(k),k-1} &:= \sum_{j=1}^s P_{j,k-1} p_{\theta(k)} j \\ \begin{bmatrix} L_{\theta(k),k} \\ K_{\theta(k),k} \\ P_{\theta(k),k} \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -I \\ 0 & 0 & 0 \\ 0 & 0 & \hat{F}_{\theta(k)} \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}^T \begin{bmatrix} 0 & I & 0 & 0 & 0 & 0 & 0 & I & 0 \\ I & -\Psi_{\theta(k),k-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & I & -R_{\theta(k)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -Q_{\theta(k)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Sigma_{\theta(k)} & \hat{I}_{\theta(k)} & -\hat{G}_{\theta(k)} \\ I & 0 & 0 & 0 & 0 & 0 & 0 & \hat{I}_{\theta(k)}^T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\hat{G}_{\theta(k)}^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

Step 2: For $k = 1$ and $\theta(k) = i \in \mathcal{S}$, make:

$$\begin{bmatrix} x_{k+1}^* \\ u_k^* \end{bmatrix} = \begin{bmatrix} L_{\theta(k),k} \\ K_{\theta(k),k} \end{bmatrix} x_k$$

Repeat Steps 1 and 2 for all $k = 2, \dots, N$.

3.4.6 Mode-independent (PM-LQR)

The Robust Linear-Quadratic regulator for discrete-time Markovian jump linear systems subject to uncertainties, unknown mode and unknown probability transition matrix (MLSUM) can be obtained by the presented below considering notation as presented by (BORTOLIN, 2017).

Consider the following MLSUM

$$\begin{cases} x_{k+1} = (F_{\theta(k)} + \delta F_{\theta(k)})x_k + (G_{\theta(k)} + \delta G_{\theta(k)})u_k, & \forall k \geq 0 \\ y_k = \mathcal{C}_{\theta(k)}x_k + \mathcal{D}_{\theta(k)}u_k, & \theta(0) \sim \pi_0 \end{cases} \quad (3.103)$$

where $(x_k, \theta(k))$ is the system state, $u_k \in \mathbb{R}^m$ is the control input and y_k is the system observation variable. $F_{\theta(k)} \in \mathbb{R}^{n \times n}$, $G_{\theta(k)} \in \mathbb{R}^{n \times m}$, $\mathcal{C}_{\theta(k)} \in \mathbb{R}^{r \times n}$ and $\mathcal{D}_{\theta(k)} \in \mathbb{R}^{r \times m}$ are known nominal system matrices; and $\delta F_{\theta(k)} \in \mathbb{R}^{n \times n}$ and $\delta G_{\theta(k)} \in \mathbb{R}^{n \times m}$ are uncertainty matrices modelled by

$$\begin{bmatrix} \delta F_{\theta(k)} & \delta G_{\theta(k)} \end{bmatrix} = H_{\theta(k)} \Delta_{\theta(k)} \begin{bmatrix} E_{F_{\theta(k)}} & E_{G_{\theta(k)}} \end{bmatrix}, \quad \forall k \geq 0, \quad (3.104)$$

with $H_{\theta(k)} \in \mathbb{R}^{n \times t}$ (non null matrix); $E_{F_{\theta(k)}} \in \mathbb{R}^{l \times n}$ and $E_{G_{\theta(k)}} \in \mathbb{R}^{l \times m}$ known; and $\Delta_{\theta(k)} \in \mathbb{R}^{t \times l}$ such that $\|\Delta_{\theta(k)}\| \leq 1$. The Markov chain is indexed by $\theta(k) \in \mathcal{S} := \{1, \dots, s\}$ related to a transition probability matrix \mathbb{P}^δ defined in Section 3.4.5.1.

The purpose is to design an optimal feedback control sequence $\{u_k^*\}_{k=0}^{N-1}$ that regulates the MLSUM (3.103), based on a control law $u_k = K_k x_k$, for $K \in \mathbb{R}^{m \times n}$ and $\forall k \geq 0$.

One can define the optimization problem for this regulator class in the following manner:

$$\min_{x_{k+1}, u_k} \max_{\delta} \mathfrak{J}_{\mu, k}(x_{k+1}, u_k), \quad (3.105)$$

for all $k = N - 1, \dots, 0$, $\delta = \{\delta_{\mathcal{Q}_k}, \delta_{\mathcal{R}_k}, \delta E_k, \delta \mathcal{A}_k, \delta \mathcal{B}_k\}$, and where the penalized regularized quadratic cost function $\mathfrak{J}_{\mu, k}(\cdot)$ is defined by (3.106)

$$\begin{aligned} \mathfrak{J}_{\mu, k}(x_{k+1}, u_k) &= x_{k+1}^T \mathcal{P}_{k+1} x_{k+1} + \\ &\left(\begin{bmatrix} 0 & \delta_{\mathcal{R}_k} \\ 0 & 0 \\ \delta E_k & -\delta \mathcal{B}_k \end{bmatrix} \begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} - \begin{bmatrix} 0 \\ -\delta_{\mathcal{Q}_k} \\ \delta \mathcal{A}_k \end{bmatrix} x_k \right)^T \begin{bmatrix} \mathcal{R} & 0 & 0 \\ 0 & \mathcal{Q} & 0 \\ 0 & 0 & \mu I_n \end{bmatrix} (\bullet) \end{aligned} \quad (3.106)$$

where $\mathcal{P}_{k+1} \succ 0$, $\mathcal{R} \succ 0$ and $\mathcal{Q} \succ 0$. Moreover,

$$\delta_{\mathcal{R}_k} = H_{\mathcal{R}} \Delta_{\mathcal{R}_k} E_{\mathcal{R}}, \quad (3.107)$$

with $H_{\mathcal{R}} = I_{sm}$, $E_{\mathcal{R}} = I_m$ and $\Delta_{\mathcal{R}_k} = [\|_{\{\theta(k)=1\}} I_m \|_{\{\theta(k)=2\}} I_m \dots \|_{\{\theta(k)=s\}} I_m]$;

$$\delta_{\mathcal{Q}_k} = H_{\mathcal{Q}} \Delta_{\mathcal{Q}_k} E_{\mathcal{Q}}, \quad (3.108)$$

with $H_{\mathcal{Q}} = I_{sn}$, $E_{\mathcal{Q}} = I_n$ and $\Delta_{\mathcal{Q}_k} = [\|_{\{\theta(k)=1\}} I_n \|_{\{\theta(k)=2\}} I_n \dots \|_{\{\theta(k)=s\}} I_n]$;

δE_k , $\delta \mathcal{A}_k$ and $\delta \mathcal{B}_k$ is defined as the Singular Augmented System

$$\delta E_k x_{k+1} = \delta \mathcal{A}_k x_k + \delta \mathcal{B}_k u_k, \quad (3.109)$$

with uncertainty matrices modelled as

$$[\delta E_k \ \delta \mathcal{A}_k \ \delta \mathcal{B}_k] = H_{\mathcal{M}} \Delta_{\mathcal{M}_k} [E_E \ E_{\mathcal{A}} \ E_{\mathcal{B}}], \quad (3.110)$$

where

$$\begin{aligned} H_{\mathcal{M}} &= \begin{bmatrix} \mathbf{h}_{11} & \dots & 0 & \mathbf{h}_{21} & \dots & 0 & \dots & \mathbf{h}_{s1} & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{h}_{1s} & 0 & \dots & \mathbf{h}_{2s} & \dots & 0 & \dots & \mathbf{h}_{ss} \end{bmatrix} \\ E_E &= [\mathbf{e}_{11} \dots \mathbf{e}_{1s} \dots \mathbf{e}_{s1} \dots \mathbf{e}_{ss}]^T \\ E_{\mathcal{A}} &= [\mathbf{a}_{11} \dots \mathbf{a}_{1s} \dots \mathbf{a}_{s1} \dots \mathbf{a}_{ss}]^T \\ E_{\mathcal{B}} &= [\mathbf{b}_{11} \dots \mathbf{b}_{1s} \dots \mathbf{b}_{s1} \dots \mathbf{b}_{ss}]^T \\ \Delta_{\mathcal{M}_k} &= \text{diag}(\Delta_{\mathcal{M},11}, \dots, \Delta_{\mathcal{M},1s}, \dots, \Delta_{\mathcal{M},s1}, \dots, \Delta_{\mathcal{M},ss}). \end{aligned} \quad (3.111)$$

Individual elements are defined based on whether the system is uncertain and the transition matrix is unknown or not. Therefore, consider $H_{\mathcal{M}} \in \mathbb{R}^{sn \times 2s^2(n+t)}$, $\Delta_{\mathcal{M}_k} \in \mathbb{R}^{2s^2(n+t) \times 2s^2(n+l)}$, $E_E \in \mathbb{R}^{2s^2(n+l) \times n}$, $E_{\mathbb{A}} \in \mathbb{R}^{2s^2(n+l) \times n}$ and $E_{\mathbb{B}} \in \mathbb{R}^{2s^2(n+l) \times m}$ with elements

$$\mathbf{h}_{ij} = [I_n \ H_i \ h_{ij} I_n \ h_{ij} H_i], \ \mathbf{e}_{ij} = [p_{ij} I_n \ 0_{n,l} \ e_{ij} I_n \ 0_{n,l}], \quad (3.112)$$

$$\mathbf{a}_{ij} = [p_j F_i^T \ p_{ij} E_{F_i}^T \ e_{ij} F_i^T \ e_{ij} E_{F_i}^T], \ \mathbf{b}_{ij} = [p_j G_i^T \ p_{ij} E_{G_i}^T \ e_{ij} G_i^T \ e_{ij} E_{G_i}^T], \quad (3.113)$$

$$\text{and } \Delta_{\mathcal{M}_{ij}} = \text{diag}(\|_{\{\theta(k)=i\}} I_n, \|_{\{\theta(k)=i\}} \Delta_i, \|_{\{\theta(k)=i\}} \Delta_{ij} I_n, \|_{\{\theta(k)=i\}} \Delta_{ij} \Delta_i), \quad (3.114)$$

with p_{ij} nominal transition probabilities, h_{ij} , Δ_{ij} and e_{ij} scalars such that $\delta p_{ij} = h_{ij}\Delta_{ij}e_{ij}$, as defined in Section 3.4.5.1.

Omitting intermediary steps for brevity, the recursive robust regulator for MLSUM is presented on Table 5, where

$$\hat{E} \leftarrow \begin{bmatrix} 0 \\ 0 \\ E_E \end{bmatrix}, \hat{\mathcal{A}} \leftarrow \begin{bmatrix} 0 \\ -E_Q \\ E_A \end{bmatrix}, \hat{B} \leftarrow \begin{bmatrix} -E_R \\ 0 \\ E_B \end{bmatrix}, \Sigma \leftarrow \begin{bmatrix} \hat{\lambda}^{-1}I & 0 & 0 \\ 0 & \hat{\lambda}^{-1}I_m & 0 \\ 0 & 0 & \hat{\lambda}^{-1}I_n \end{bmatrix}, \text{ and } \hat{\lambda} > 0. \quad (3.115)$$

Table 5: Robust Regulator for MLSUM

Initial conditions: Define N , $\lambda > 0$, E_E , E_Q , E_A , E_R , E_B and $P_{\lambda,N} \succ 0$.

Step 1: (*Backwards*) Calculate for all $k = N - 1, \dots, 0$:

$$\begin{bmatrix} \mathcal{L}_{\lambda,k} \\ \mathcal{K}_{\lambda,k} \\ \mathcal{P}_{\lambda,k} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \hat{\mathcal{A}} \\ I_n & 0 & 0 \\ 0 & I_m & 0 \end{bmatrix}^T \begin{bmatrix} \mathcal{P}_{\lambda,k+1}^{-1} & 0 & I_n & 0 \\ 0 & \Sigma & \hat{E} & -\hat{\mathcal{B}} \\ I_n & \hat{E}^T & 0 & 0 \\ 0 & -\hat{\mathcal{B}}^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \hat{\mathcal{A}} \\ 0 \\ 0 \end{bmatrix}$$

Step 2: (*Forwards*) For each $k = 0, \dots, N - 1$, make:

$$\begin{bmatrix} x_{k+1}^* \\ u_k^* \end{bmatrix} = \begin{bmatrix} L_{\lambda,k} \\ K_{\lambda,k} \end{bmatrix} x_k$$

When dealing with multirotors, the same case as in Section 3.4.5.1 applies for the PM-LQR, which means it is not possible to generate control inputs from backwards as suggested on Step 1 of Table 5. Therefore, convergence of the Riccati $P_{\lambda,k}$ is assumed when calculating control inputs as suggested in Table 6.

With the controller framework in hands, it is necessary to relate this framework to the model for multirotor control. In this case, the relation is similar to the one presented on Section 3.4.5.1. Following the same as in Section 3.4.5.1, for each failure mode there is a different $\hat{F}_{\theta(k)}$ and $\hat{G}_{\theta(k)}$, and all these matrices are used together to provide a unique $P_{\lambda,k}$. The difference is there is no FDD to identify at which mode the system is at.

Let $\Lambda_{\theta(k)}$ and $A_{\theta(k)}$ be the same as in Equations (3.46) and (3.83), with the difference that $\theta(k)$ represent a failure mode from a set \mathcal{S} of previously defined failure modes. Hence,

Table 6: Robust Regulator for MLSUM considering only forwards calculation

Initial conditions: Define N , $\lambda > 0$, E_E , E_Q , E_A , E_R , E_B and $P_{\lambda,0} \succ 0$.

Step 1: Calculate for $k = 1$:

$$\begin{bmatrix} \mathcal{L}_{\lambda,k} \\ \mathcal{K}_{\lambda,k} \\ \mathcal{P}_{\lambda,k} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \hat{\mathcal{A}} \\ I_n & 0 & 0 \\ 0 & I_m & 0 \end{bmatrix}^T \begin{bmatrix} \mathcal{P}_{\lambda,k-1}^{-1} & 0 & I_n & 0 \\ 0 & \Sigma & \hat{E} & -\hat{\mathcal{B}} \\ I_n & \hat{E}^T & 0 & 0 \\ 0 & -\hat{\mathcal{B}}^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \hat{\mathcal{A}} \\ 0 \\ 0 \end{bmatrix}$$

Step 2: Make:

$$\begin{bmatrix} x_{k+1}^* \\ u_k^* \end{bmatrix} = \begin{bmatrix} L_{\lambda,k} \\ K_{\lambda,k} \end{bmatrix} x_k$$

Repeat Steps 1 and 2 for all $k = 2, \dots, N$.

the multirotor control can be obtained by making

$$F_{\theta(k)} \leftarrow c2d \left(\begin{bmatrix} A_{\theta(k)}(\boldsymbol{\omega}_B, \omega_i) & 0_{3x3} \\ \frac{1}{2}S(q)_{(2:4,:)} & 0_{3x3} \end{bmatrix}, \delta t \right) \quad G_{\theta(k)} \leftarrow c2d \left(\begin{bmatrix} -I_{cg}^{-1}M_t\Lambda_{\theta(k)} \\ 0_{3x3} \end{bmatrix}, \delta t \right) \quad (3.116)$$

$$x_k \leftarrow \begin{bmatrix} \boldsymbol{\omega}_B^d - \boldsymbol{\omega}_B \\ \mathbf{q}_e \end{bmatrix} - I_{cg}^{-1} \left(-I_{cg}^{-1}M_t u_k^* - \dot{\boldsymbol{\omega}}_B^d + A(\boldsymbol{\omega}_B, \omega_i)\boldsymbol{\omega}_B^d \right) \rightarrow \boldsymbol{\Gamma}_B^d.$$

M_f and $A(\boldsymbol{\omega}_B, \omega_i)$ are the nominal cases in the Equation for $\boldsymbol{\Gamma}_B^d$.

(BORTOLIN, 2017) remarks that the application of the above mentioned algorithm is conditioned to $\|\Delta\| \leq 1$, as presented above. When this condition is not satisfied, a scalar $\kappa \geq 1$ shall be considered, such that $\|\Delta\| \leq \kappa$ and following the equivalence

$$\Delta_{\mathcal{M}_k} \leftarrow \frac{\Delta_{\mathcal{M}_k}}{\kappa}, \quad \Delta_{\mathcal{R}} \leftarrow \frac{\Delta_{\mathcal{R}}}{\kappa}, \quad \Delta_{\mathcal{Q}} \leftarrow \frac{\Delta_{\mathcal{Q}}}{\kappa}$$

$$E_E \leftarrow \kappa E_E, \quad E_A \leftarrow \kappa E_A \quad E_B \leftarrow \kappa E_B \quad E_R \leftarrow \kappa E_R \quad E_Q \leftarrow \kappa E_Q. \quad (3.117)$$

3.5 Control Allocation

For the control architectures that do not generate rotor speeds directly, it is necessary to distribute the virtual control inputs $\boldsymbol{\Gamma}_B^d$ and \boldsymbol{T}_A^d (or \boldsymbol{T}_A^{d*}) to rotor speeds. Following the same nomenclature of the attitude controllers presented on Section 3.4, the control allocation algorithms presented here can also be classified in "active" or "passive" when it comes to online adaptation to rotor failures or not.

Section 3.5.1 presents a simple control allocation method based on the Moore-Penrose Pseudo-Inverse (BARATA; HUSSEIN, 2012). Section 3.5.2 introduces the Redistributed Pseudo-Inverse method proposed to deal with actuator saturation. Section 3.5.3

proposes an extension of the method presented on ([MICHIELETTTO; RYLL; FRANCHI, 2017](#)) that uses the null-space of the torque matrix M_t to provide better results. Finally, Sections [3.5.1.1](#), [3.5.2.1](#) and [3.5.3.1](#) present the "active" cases for the control allocations cited above.

3.5.1 Pseudo-Inverse

According to Equations [\(3.8\)](#) and [\(3.43\)](#), one can consider

$$\begin{bmatrix} Q^{-1}\mathbf{T}_A^d \\ \mathbf{\Gamma}_B^d \end{bmatrix} = \begin{bmatrix} M_f \\ M_t \end{bmatrix} \boldsymbol{\Omega} = H\boldsymbol{\Omega} \quad (3.118)$$

to be true. According to ([JOHANSEN; FOSSEN, 2013](#)), as long as H is full column rank, rotor speeds can be obtained by making

$$\boldsymbol{\Omega}^d = H^\dagger \begin{bmatrix} Q^{-1}\mathbf{T}_A^d \\ \mathbf{\Gamma}_B^d \end{bmatrix}, \quad (3.119)$$

$$H^\dagger = H^T(HH^T)^{-1}, \quad (3.120)$$

where H^\dagger is the Moore-Penrose Pseudo-Inverse. Taking the square root of the elements of $\boldsymbol{\Omega}^d$ and recasting the original rotor rotation directions, desired rotor speeds can be obtained by

$$\boldsymbol{\omega}^d = [d_1\omega_1^d, \dots, d_N\omega_N^d]^T. \quad (3.121)$$

3.5.1.1 Active Pseudo-Inverse

For the case where rotor failures are considered in the control allocation, one can make ([ALWI; EDWARDS, 2015](#))

$$H_{fdd} = \begin{bmatrix} M_f\Lambda_{fdd} \\ M_t\Lambda_{fdd} \end{bmatrix}, \quad (3.122)$$

following the same definition of Λ_{fdd} as in Section [3.4.2.6](#). In cases when H or H_{fdd} is not full column rank, ([JOHANSEN; FOSSEN, 2013](#)) proposes

$$H^\dagger = H^T(HH^T + \varepsilon I)^{-1}, \quad (3.123)$$

where $\varepsilon > 0$ is a small regulation parameter.

3.5.2 Redistributed Pseudo-Inverse

The Redistributed Pseudo-Inverse method, as presented by ([MARKS; WHID-BORNE; YAMAMOTO, 2012](#)) is an extension of the Pseudo-Inverse capable of dealing with actuator saturation. Although this is not the only method to deal with actuator restrictions, as pointed out on Section [1.1](#), it offers a simple and relatively robust solution to the problem.

Let ω_i^{max} and ω_i^{min} be the maximum and minimum rotor speeds for each rotor i , respectively, with $i = 1, \dots, m$ and $\boldsymbol{\omega} \in \mathbb{R}^m$. Define H according to Equation (3.118). Define $c \in \mathbb{R}^m$ such that $c = [c_1, \dots, c_m]^T$. The redistributed allocation is obtained by Algorithm 1.

Algorithm 1 Redistributed Pseudo-Inverse Control Allocation

- 1: Make $\widetilde{H} = H$ and $c_i = 0, \forall i$.
 - 2: Solve $\boldsymbol{\Omega}^d = -c + \widetilde{H}^\dagger \left(\begin{bmatrix} Q^{-1} \mathbf{T}_A^d \\ \mathbf{\Gamma}_B^d \end{bmatrix} + Hc \right)$.
 - 3: If $\omega_i^{min} \leq \omega_i \leq \omega_i^{max}, \forall i$, then finish. Otherwise, for $i = 1, \dots, m$,
 - If $\omega_i \leq \omega_i^{min}$, make $c_i = -(\omega_i^{min})^2$, remove column i from \widetilde{H}
 - If $\omega_i \geq \omega_i^{max}$, make $c_i = -(\omega_i^{max})^2$, remove column i from \widetilde{H}
and repeat Step 2.
-

Taking the square root of the elements of $\boldsymbol{\Omega}$ and recasting the original rotor rotation directions, desired rotor speeds can be obtained by Equation (3.121).

3.5.2.1 Active Redistributed Pseudo-Inverse

In the case of rotor failures, one can make the equivalence

$$H \leftarrow H_{fdd}, \quad (3.124)$$

$$\omega_i^{min} \leftarrow \xi_{i,fdd} \omega_i^{min} \quad \text{and} \quad (3.125)$$

$$\omega_i^{max} \leftarrow \xi_{i,fdd} \omega_i^{max} \quad (3.126)$$

for the algorithm presented on Section 3.5.2 and $\xi_{i,fdd}$ as presented on Section 4.1.6.

3.5.3 Null-space based control allocation

For the high-level control to command both $\mathbf{\Gamma}_B^d$ and \mathbf{T}^d , the multirotor has to have full-actuation over the rotation dynamics, which means $\text{rank}(M_t) \geq 3$. Moreover, at least $|\mathbf{T}^d|$ has to be generated independently of $\mathbf{\Gamma}_B^d$, in the case of sub-actuated multirotors for the algorithm in Section 3.3 to be feasible. Therefore, $\text{rank}([M_f^T M_t^T]) \geq 4$. The latter statement means that $\exists \mathbf{b}_2 \in \ker(M_t) : M_f \mathbf{b}_2 \neq 0$, following notation \mathbf{b}_2 from (MICHELETTI; RYLL; FRANCHI, 2017). If this is true, the command allocation can be the composition between rotor speeds to generate $|\mathbf{T}^d|$ (or \mathbf{T}^d in case of fully-actuated multirotors) and rotor speeds to generate $\mathbf{\Gamma}_B^d$. Hence,

$$\begin{aligned} \exists \boldsymbol{\Omega}^d = \boldsymbol{\Omega}_f + \boldsymbol{\Omega}_\tau \in \mathbb{R}^N : |\mathbf{T}_B^d| &= |M_f \boldsymbol{\Omega}^d| \text{ and} \\ \mathbf{\Gamma}_B^d &= M_t \boldsymbol{\Omega}^d, \end{aligned} \quad (3.127)$$

where

$$\boldsymbol{\Omega}_f = \mathbf{b}_2 c, \quad (3.128)$$

$$\boldsymbol{\Omega}_\tau = M_t^\dagger \mathbf{\Gamma}_B^d. \quad (3.129)$$

(BARATA; HUSSEIN, 2012) proves that the Moore-Penrose Pseudo-Inverse solution for Ω_τ is unique, always exists, and is the closest one to Γ_B^d in case it is not exactly the same.

(MICHELETTI; RYLL; FRANCHI, 2017) suggests to use the gravity direction as reference for choosing \mathbf{b}_2 indirectly. The author's approach, however, does not provide a final solution, leaving the choice of \mathbf{b}_2 open to the reader. This present work proposes an alternative to choose \mathbf{b}_2 by weighting between maneuverability and the ability of the multirotor to generate \mathbf{T}_B^d at an arbitrary attitude even when \mathbf{T}_B^d is not aligned with \hat{k} .

The case of best maneuverability is defined here as the case which each rotor spin in the middle of the range of allowable speeds for it. This choice forces the actuators to run outside the non-linear region of reverse rotation, solving a problem present on the allocations presented on Sections 3.5.1 and 3.5.2. This difference will be clarified on Sections 4.2 and 5.1. Let one define a cost function

$$\begin{aligned} J = & \| \ker(M_t)\mathbf{x} - \Omega_{1/2} \|_{W_m} \\ & + \| M_f \ker(M_t)\mathbf{x} - \mathbf{T}_B^d \|_{W_a}, \end{aligned} \quad (3.130)$$

where $\ker(M_t) = N_s$ is a matrix representation of the null-space of M_t and

$$\Omega_{1/2} = \left[\frac{(\omega_1^{max})^2 + (\omega_1^{min})^2}{2}, \dots, \frac{(\omega_N^{max})^2 + (\omega_N^{min})^2}{2} \right]^T,$$

where $W_m \in \mathbb{R}$ is the maneuverability weight and $W_a \in \mathbb{R}$ is the "attitude following" weight.

Vector \mathbf{b}_2 can be chosen by making

$$\mathbf{b}_2 = N_s \mathbf{x}^*, \quad (3.131)$$

where

$$\mathbf{x}^* = \text{argmin}_v(J) \quad (3.132)$$

$$\begin{aligned} &= \left[N_s^T (W_m N_s + M_f^T W_a M_f N_s) \right]^{-1} \bullet, \\ &\bullet \left[N_s^T (W_m \Omega_{1/2} + M_f^T W_a \mathbf{T}_B^d) \right], \end{aligned} \quad (3.133)$$

\mathbf{b}_2 represents, therefore, a direction in the \mathbb{R}^N space of rotor speeds capable of following \mathbf{T}_B^d or providing the best maneuverability according to weights W_a and W_m .

At last, it is necessary to make $|M_f \Omega|$ as close as possible to $|\mathbf{T}_B^d|$. This is achieved using

$$c = (\mathbf{T}_B^d - M_f \Omega_\tau)^T \frac{M_f \mathbf{b}_2}{\| M_f \mathbf{b}_2 \|^2}, \quad (3.134)$$

where

$$c \in \mathbb{R} : c = \text{argmin}_c \| M_f (\Omega_\tau + \mathbf{b}_2 c) - \mathbf{T}_B^d \|^2. \quad (3.135)$$

In case W_a weight requires rotor speeds that are not feasible, they need to pass through the transformation of Equation (3.10), resulting in Ω' . Moreover, it is necessary to recast original rotor rotation directions. Therefore, the control allocation results in the following rotor speeds to be passed to actuator-level control:

$$\boldsymbol{\omega}^d = [d_1\omega'_1, \dots, d_N\omega'_N]^T. \quad (3.136)$$

3.5.3.1 Active Null-space based control allocation

In the case of online rotor failure updating by an FDD, one can make

$$M_f \leftarrow M_f \Lambda_{fdd}, \quad (3.137)$$

$$M_t \leftarrow M_t \Lambda_{fdd}, \quad (3.138)$$

$$\omega_i^{min} \leftarrow \xi_{i,fdd} \omega_i^{min} \quad \text{and} \quad (3.139)$$

$$\omega_i^{max} \leftarrow \xi_{i,fdd} \omega_i^{max} \quad (3.140)$$

over the algorithm of Section 3.5.3.

4 MATERIALS AND METHODS

The main purpose of this work is to compare diverse controllers in multirotor control when it comes to their robustness to failures and parameter variations. For this comparison, a common multirotor architecture was chosen for all control architectures so that it could be simulated and implemented in experiments. The chosen multirotor was the coaxial octarotor because of its higher lift-to-weight ratio ([RINALDI; GARGIOLI; QUAGLIOTTI, 2014](#)) and its actuator redundancy for rotor failure study.

All simulations were executed in a Matlab environment with the help of a parallel computing cluster. For experimental results, a large size coaxial octacopter was constructed. This work was able to finish only the first flight tests with default auto-pilot controllers, leaving the implementation of the controllers herein compared to future works. The octacopter construction and its characteristics are described on Appendix A. For simulated results, two octacopter models were considered: an idealized octacopter based on values taken from ([MERHEB; BATEMAN; NOURA, 2015](#)); and an estimated model for the constructed octarotor based on a CAD model of it, as presented on Figure 5. They shall be referenced as Model 1 and Model 2, respectively. Tables 7 and 8 present some parameters for each model, respectively.

Table 7: Model 1 parameters

Parameter	Value
\mathbf{r}_1	$[0.283 \ 0.283 \ 0.050]^T m$
\mathbf{r}_2	$[-0.283 \ 0.283 \ 0.050]^T m$
\mathbf{r}_3	$[-0.283 \ -0.283 \ 0.050]^T m$
\mathbf{r}_4	$[0.283 \ -0.283 \ 0.050]^T m$
\mathbf{r}_5	$[0.283 \ 0.283 \ -0.050]^T m$
\mathbf{r}_6	$[-0.283 \ 0.283 \ -0.050]^T m$
\mathbf{r}_7	$[-0.283 \ -0.283 \ -0.050]^T m$
\mathbf{r}_8	$[0.283 \ -0.283 \ -0.050]^T m$
I_{cg}	$diag([44e^{-3} \ 44^{-3} \ 88^{-3}]) \ kg.m^2$
μ_B	$diag([0.25 \ 0.25 \ 0.25]) \ kg.s^{-1}$
$\mathbf{n}_1, \dots, \mathbf{n}_8$	$[0 \ 0 \ 1]^T$
m	$1.64 \ kg$
C_{L1}, \dots, C_{L8}	$10^{-6} \ Ns^2$
C_{D1}, \dots, C_{D8}	$0.3 \ 10^{-6} \ Nms^2$
I_1, \dots, I_8	$90 \ 10^{-6} \ kg.m^2$
$[d_1, \dots, d_8]$	$[1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ 1]^T$
$\omega_1^{max}, \dots, \omega_8^{max}$	$1500 \ rad/s$
$\omega_1^{min}, \dots, \omega_8^{min}$	$0 \ rad/s$



Figure 5: 3D CAD representation of Model 2.

Section 4.1 describes the multirotor simulator developed in Matlab as a tool for control development and analysis. Only one control allocation method from Section 3.5 is used in the attitude controller comparisons so as to reduce the number of variables. Therefore, Section 4.2 presents the comparison method between these control allocation algorithms in order to choose the most viable one. Section 4.3 deepens the control allocation analysis for the chosen algorithm and shows its features and advantages. Considering the diversity of controllers to be compared, an optimization method was needed to tune each controller removing the most subjectivity from this process as possible. Section 4.4 describes this tuning algorithm. Lastly, Section 4.5 presents the evaluation method for each controller so as to provide a means of comparison between all architectures.

The attitude planning algorithm presented on Section 3.3 is present in all simulations, being therefore tested on every test case. The method on Section 4.3, however, also serves as a validation method for the attitude planning algorithm of Section 3.3, showing its robustness to different multirotor architectures.

Table 8: Model 2 parameters

Parameter	Value
\mathbf{r}_1	$[0.344 \ 0.343 \ 0.014]^T m$
\mathbf{r}_2	$[-0.341 \ 0.342 \ 0.014]^T m$
\mathbf{r}_3	$[-0.341 \ -0.343 \ 0.014]^T m$
\mathbf{r}_4	$[0.344 \ -0.342 \ 0.014]^T m$
\mathbf{r}_5	$[0.334 \ 0.338 \ 0.091]^T m$
\mathbf{r}_6	$[-0.336 \ 0.337 \ 0.091]^T m$
\mathbf{r}_7	$[-0.336 \ -0.338 \ 0.091]^T m$
\mathbf{r}_8	$[0.339 \ -0.338 \ 0.091]^T m$
\mathbf{n}_1	$[-0.062 \ -0.062 \ 0.996]^T m$
\mathbf{n}_2	$[0.062 \ -0.062 \ 0.996]^T m$
\mathbf{n}_3	$[0.062 \ 0.062 \ 0.996]^T m$
\mathbf{n}_4	$[-0.062 \ 0.062 \ 0.996]^T m$
\mathbf{n}_5	$[-0.062 \ -0.062 \ 0.996]^T m$
\mathbf{n}_6	$[0.062 \ -0.062 \ 0.996]^T m$
\mathbf{n}_7	$[0.062 \ 0.062 \ 0.996]^T m$
\mathbf{n}_8	$[-0.062 \ 0.062 \ 0.996]^T m$
I_{cg}	$\begin{bmatrix} 0.3143978800 & 0.0000861200 & -0.0014397600 \\ 0.0000861200 & 0.3122127800 & 0.0002368800 \\ -0.0014397600 & 0.0002368800 & 0.5557912400 \end{bmatrix} kg.m^2$
μ_B	$diag([0.25 \ 0.25 \ 0.25]) \ kg.s^{-1}$
m	$6.015 \ kg$
C_{L1}, \dots, C_{L8}	Section 4.1.4
C_{D1}, \dots, C_{D8}	Section 4.1.4
I_1, \dots, I_8	$479.35^{-6} \ kg.m^2$
$[d_1, \dots, d_8]$	$[1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ 1]^T$
$\omega_1^{max}, \dots, \omega_8^{max}$	$729 \ rad/s$
$\omega_1^{min}, \dots, \omega_8^{min}$	$0 \ rad/s$
R_1, \dots, R_8	$0.0975 \ \Omega$
$\kappa_{\tau 1}, \dots, \kappa_{\tau 8}$	$0.02498 \ Nm/A$
$\kappa_{v1}, \dots, \kappa_{v8}$	$0.02809 \ Vs/rad$
$v_{LL1}^{max}, \dots, v_{LL8}^{max}$	$22 \ V$

4.1 Simulation framework

Two Matlab classes were implemented to provide a simulation framework for this work. The first one is the @multicopter class. It is responsible for simulating the rigid-body dynamics presented on Chapter 2 and for providing an easy simulation interface. The function *model* is responsible for encapsulating the multirotor model, making it possible to change model considerations without having to change the entire class. Some of the effects considered for this version are:

- Arbitrary number of rotors.

- Arbitrary rotor positions.
- Arbitrary rotor orientations.
- Body frame centered at CG.
- Payload changes multirotor mass, inertia tensor and CG position
- Relative or absolute parameter errors.
- No blade-flapping.
- Quaternion modelling allowing any kind of rotations.
- Only translational aerodynamic drag.
- Lift and drag proportional to square of rotor speed.
- Rotor failure types: Rotor stuck, non-responding, motor failure, propeller failure.
- Rotors variables can be changed during run time, but they are considered as constants and not dynamic variables in the model.
- Euler angles are [row, pitch, yaw] always in this order, assuming a (1,2,3) rotation sequence, or in other words, the nautical and aeronautical convention ([DIEBEL, 2006](#)).
- Motor models can be either a perfect translation from desired rotor speeds to real rotor speeds; a second-order transfer function; or a brushless model based on electromagnetic motor coefficients.
- Rotors are not revertible.
- Disturbance is a linear 3D force.

The second simulation class is the `@multicontrol`. It is a child class from `@multicopter` to implement the presented on Chapter [3](#), encapsulating all functions related to

- Control allocators and its configurations,
- Attitude planning algorithms and its configurations,
- Fault detection and diagnostics simulation,
- Trajectory definition,
- Simulation metrics,
- Simulation plotting

and others. The simulation classes, results, methods and all files can be obtained at [<https://github.com/lbf10/multirotor>](https://github.com/lbf10/multirotor). Appendix B presents an example of a code to simulate the octarotor with a few lines using the presented classes.

4.1.1 Simulation solver

Considering the model on Chapter 2, diverse differential equations have to be solved. Matlab provides built-in ODE solvers optimized for stiff or non-stiff differential equations([MATHWORKS, 2019e](#)). @multicopter class allows four ODE solving methods to be used: two built-in Matlab solvers, ODE45 ([MATHWORKS, 2019d](#)) and ODE15S ([MATHWORKS, 2019c](#)); and two classic solving methods, the Euler method ([HERMANN; SARAVI, 2014](#)) and the Midpoint method, which is the modified Euler method.

ODE45 and ODE15S are optimized algorithms with variable time steps. They guarantee the solution to be within a chosen error at each time step, improving accuracy, but take much more time to solve than classic methods like the Euler and Midpoint. The Midpoint method has double the accuracy of the Euler method, but it takes more time in order to perform double the number of function evaluations used in Euler's.

As will be presented on Section 4.4, a large number of simulations had to be generated, making it necessary to compromise between accuracy and simulation time. Therefore, for Section 4.4 the Euler solver was chosen with a simulation time step ten times smaller than the control loop period, but not greater than 5 ms. On the other hand, ODE45 solver was chosen for the controllers evaluation on Section 4.5, with simulation time step of 2 ms.

It is important to note that the simulation time step has two meanings. In the case of the classic solvers, the simulation time step is the same as the solver's time step. In the second case of ODE45 or ODE15S, the simulation time step defines the times at which the solver shall evaluate the differential equations, and not the solver's time step, which is variable and guarantee the minimum accuracy.

4.1.2 Control loop delay

For every control iteration, a control loop period is considered. The simulation sampling period δt_{sim} is always at least 10 times smaller than the control loop period δt . The control actuation is not necessarily at the beginning of the control loop. A control delay is considered from the time when the control loop starts, reading all multirotor states, to the time the actuators are inputed. For all the optimization simulations in Section 4.4, the control delay was considered to be of 20% of the entire control loop period.

4.1.3 Desired angular speeds and accelerations

On all Sections of Chapter 3, the desired angular velocities and angular accelerations ω_B^d and $\dot{\omega}_B^d$ were assumed as given values. In the simulations of the @multicopter class presented here, only the current body attitude q and attitude reference q_{ref} , therefore the attitude error q_e , and the current body angular velocity ω_B are provided. To calculate ω_B^d and $\dot{\omega}_B^d$ one can make

$$\omega_B^d(t) = \alpha \omega_B^d(t - \delta t) + (1_{3x3} - \alpha) \frac{q_e}{\delta t} \quad (4.1)$$

$$\dot{\omega}_B^d = \frac{\omega_B^d(t) - \omega_B^d(t - \delta t)}{\delta t} \quad (4.2)$$

where $\alpha \in \mathbb{R}^{3 \times 3}$ is diagonal matrix that implements a low-pass filter on the desired angular velocity and δt is the control loop period.

4.1.4 Drag and lift coefficients

Drag and Lift coefficients may be considered constant for a range of rotor speeds. However, in general they vary with many factors, including the rotor speed itself. Simulation classes presented here allow consideration of two types of drag and lift coefficients: constant or variable in relation to rotor speeds. For the second case, it is possible to provide a dataset of coefficients per rotor speed for the model to fit the expected rotor coefficient based on a chosen method and the current rotor speed. The available fitting methods are:

- poly1: Linear polynomial curve
- poly2: Quadratic polynomial curve
- linearinterp: Piecewise linear interpolation
- cubicinterp: Piecewise cubic interpolation
- smoothingspline: Smoothing spline (curve)

Motor manufacturer of Model 2 provides a motor characterization table based on bench testing, presented on Appendix C. (KATZ, 2016) identified some motor parameters that, along with manufacturer data and assuming a reduction of 20% in the lift coefficient because of the coaxial architecture, give the lift and drag coefficients presented on Figures 6 and 7 for Model 2. In the case of simulations with constant coefficients, the selected points on Figures 6 and 7 were chosen as operational points $C_{Li_{op}}$ and $C_{Di_{op}}$. These operational points are the ones used in the control allocation algorithms.

Model 1 has only constant coefficients as provided on Table 7. However, for comparing Model 1 and Model 2 robustness as will be presented on Section 4.5, the behaviour of lift and drag coefficients for Model 2 was adapted to Model 1. This was

obtained by scaling both rotor speed and coefficient axis. Figures 8 and 9 present Lift and Drag coefficients for Model 1 for both constant and variable cases.

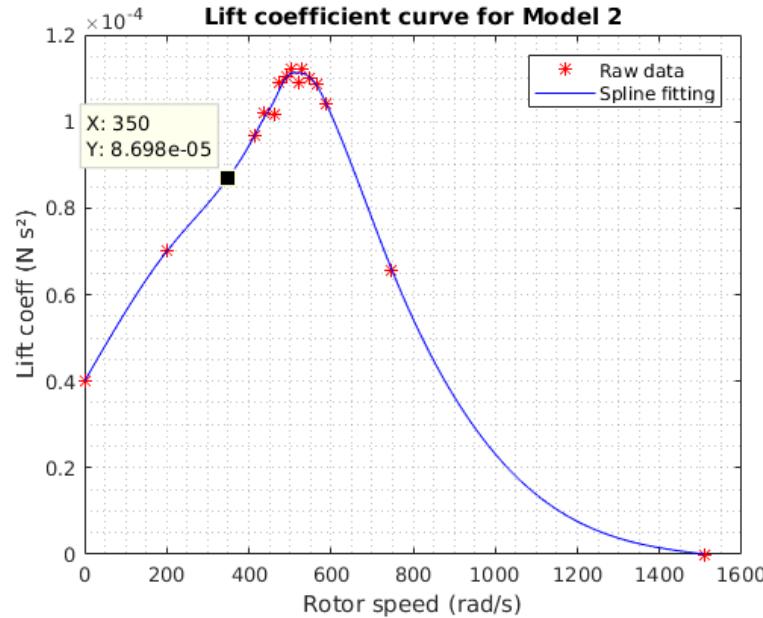


Figure 6: Fitted lift coefficient for Model 2 rotors.

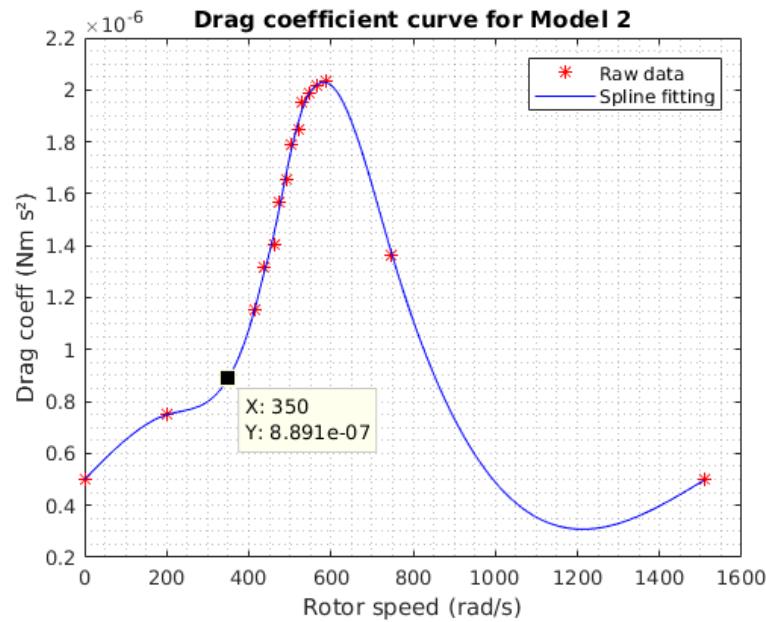


Figure 7: Fitted drag coefficient for Model 2 rotors.

4.1.5 Fault simulation

Four different rotor fault types were coded in the @multicopter class: a rotor stuck fault, a motor efficiency reduction fault, a broken propeller fault and a communication

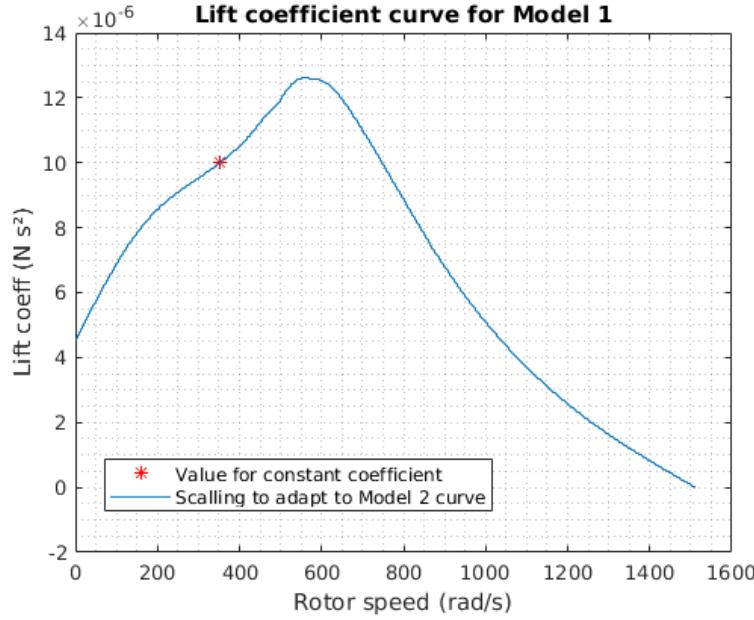


Figure 8: Lift coefficient for Model 1 rotors.

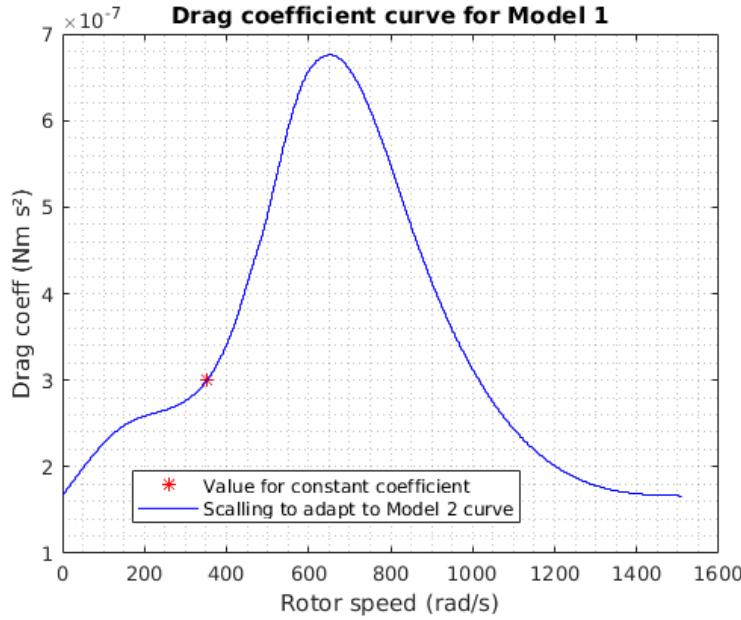


Figure 9: Drag coefficient for Model 1 rotors.

fault. In this work, only the motor efficiency reduction and broken propeller fault were considered.

Motor dynamics simulation is obtained by Equations (2.29), (2.30) and (2.31), which give

$$\dot{\omega}_i = \frac{1}{I_i} \left(\kappa_{\tau} \left(\frac{v_{LLi} - \kappa_v \omega_i}{R_i} \right) - C_{D_i} \omega_i |\omega_i| \right), \quad (4.3)$$

where the line voltage v_{LLi} is the system input.

In this work, a motor efficiency reduction is defined as a reduction in the capacity of generating torque and of rotating fast. This means making $\kappa_t \leftarrow \xi_i \kappa_t$ and $\kappa_v \leftarrow \kappa_v / \xi_i$, for $\xi_i \in \mathbb{R} \mid 0 \leq \xi_i \leq 1$ a motor efficiency coefficient. In the same sense, a propeller failure is defined as a reduction in rotor diameter, which implies, in a simple approximation, a reduction in rotor inertia, rotor drag and rotor lift. This means making

$$I_i \leftarrow \xi_{Pi}^2 I_i, \quad (4.4)$$

$$C_{Di} \leftarrow \xi_{Pi} C_{Di}, \quad (4.5)$$

and

$$C_{Li} \leftarrow \xi_{Pi} C_{Li} \quad (4.6)$$

for $\xi_{Pi} \in \mathbb{R} \mid 0 \leq \xi_{Pi} \leq 1$ a propeller efficiency coefficient. This gives the dynamic Equation (4.7) with motor response depicted on Figure 10 for a command of 22 V and a variation in motor efficiency.

$$\dot{\omega}_i = \frac{1}{I_i \xi_{Pi}^2} \left(\kappa_\tau \xi_i \left(\frac{v_{LLi} - \frac{\kappa_v \omega_i}{\xi_i}}{R_i} \right) - C_{Di} \xi_{Pi} \omega_i |\omega_i| \right), \quad (4.7)$$

If one consider that a motor failure means a reduction of only the parameter κ_t , motor

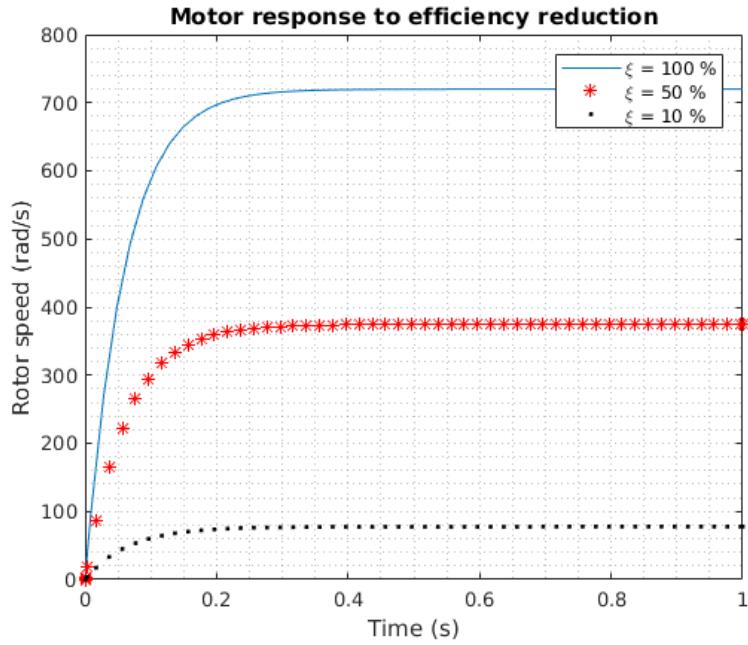


Figure 10: Motor dynamic response for model 2 considering efficiency ξ_i .

response becomes the one presented on Figure 11, which suggests that the proposition of Equation (4.7) is already the worst case scenario.

Rotor dynamics is not simulated for Model 1 as proposed by Equation (4.7). It considers, instead, a perfect mapping from ω^d to ω . In the case of a motor failure, the

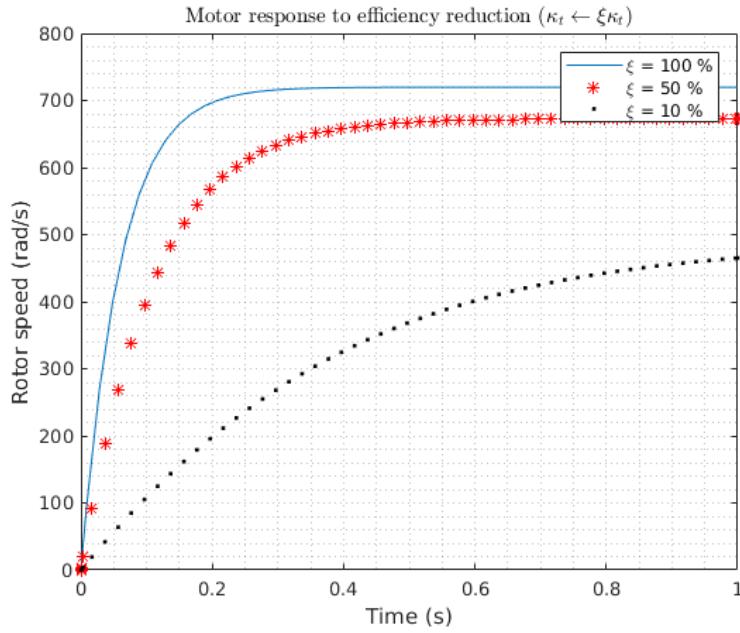


Figure 11: Motor dynamic response for model 2 considering efficiency only on the parameter κ_t .

relation

$$\boldsymbol{\omega} = \xi_i \boldsymbol{\omega}^d \quad (4.8)$$

takes place, along with Equations (4.5) and (4.6).

4.1.6 Fault Detection and Diagnosis

FDD algorithms are not part of the scope of this work. Therefore, it was necessary to simulate the response of a FDD algorithm to investigate its influence on control convergence. (QI et al., 2013) presents a review of diverse FDD methods in the literature. Based on this review, two parameters were chosen as a means to describe a FDD algorithm quality: The hit rate and the diagnostics delay.

Each rotor is considered to have its own FDD that is capable of estimating the motor and propeller efficiencies for that rotor. These estimations $\xi_{i,fdd}$ and $\xi_{P_i,fdd}$, regardless whether the rotor is in a faulty state or not, is subject to an error controlled by the hit rate in the simulation. Figure 12 shows the relation between the hit rate and $\xi_{i,fdd}$ estimation error. The simulated FDD algorithm returns an efficiency value based on the rotor current efficiency added with this random error guaranteeing values to be $0 \leq \xi_{i,fdd} \leq 1$ and $0 \leq \xi_{P_i,fdd} \leq 1$, as presented before. For this, the FDD algorithm generates two possible estimation values: one for a positive error and another for a negative error, truncating either to 0 or 1, and returning the one that gives the estimation with the largest error. The FDD delay, on the other hand, is the time between the real motor and propeller

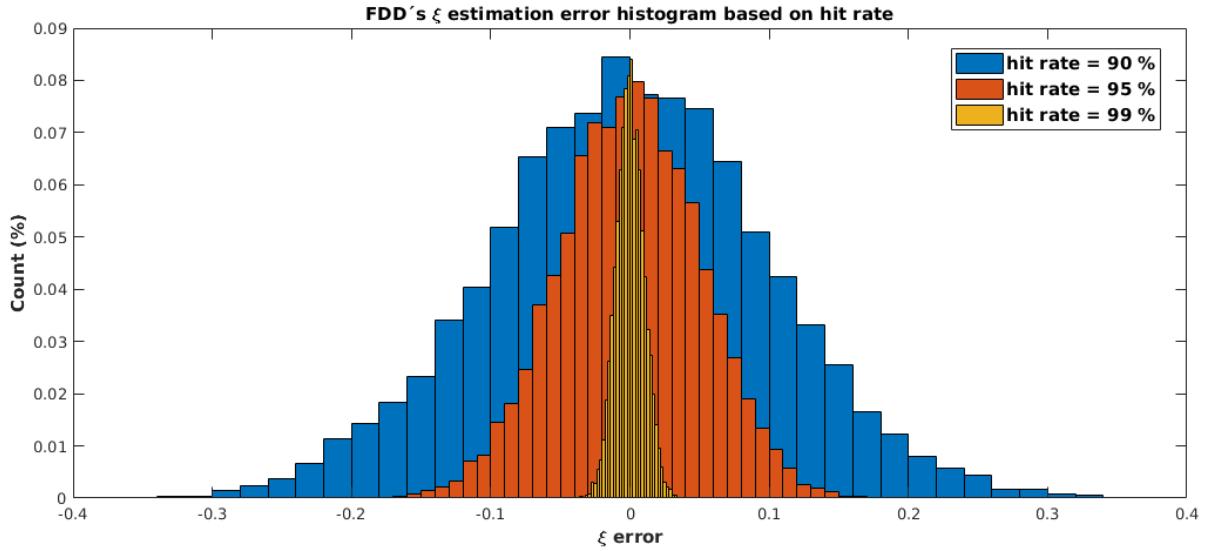


Figure 12: Histogram of the motor efficiency estimation error for the simulated FDD.

efficiencies ξ_i and ξ_{Pi} occurrences and the availability of their estimation $\xi_{i,fdd}$ and $\xi_{Pi,fdd}$ to the control algorithms by the FDD algorithm.

It is important to note that the relation between the motor health parameters and the motor and propeller efficiency parameters are the ones presented below.

$$\lambda_{Ii,fdd} := \xi_{i,fdd} \xi_{Pi,fdd}^2 \quad (4.9)$$

$$\lambda_{i,fdd} := \xi_{i,fdd}^2 \xi_{Pi,fdd} \quad (4.10)$$

4.1.7 Payload

When a payload is configured in the simulation, three parameters must be given: its relative position to the CG \mathbf{r}_{pl} ; its mass m_{pl} ; and its inertia matrix I_{pl} in relation to its CG, whose main diagonal is comprised of the main moments of inertia and the other elements are the products of inertia. According to (HIBBELE, 2016), the change in the CG position can be calculated by

$$\mathbf{r}_{cg} = \frac{m_{pl}}{m + m_{pl}}, \quad (4.11)$$

while the payload position in relation to the new CG is

$$\mathbf{r}_{pl}^* = \mathbf{r}_{pl} - \mathbf{r}_{cg}. \quad (4.12)$$

The inertia tensor for the payload and the multirotor, in relation to the new CG position, can be obtained, respectively, by

$$I_{cg}^* = \begin{bmatrix} I_{\hat{i}\hat{i}} + m(r_{cg,\hat{j}}^2 + r_{cg,\hat{k}}^2) & -I_{\hat{i}\hat{j}} - m(r_{cg,\hat{i}}r_{cg,\hat{j}}) & -I_{\hat{i}\hat{k}} - m(r_{cg,\hat{k}}r_{cg,\hat{i}}) \\ -I_{\hat{j}\hat{i}} - m(r_{cg,\hat{i}}r_{cg,\hat{j}}) & I_{\hat{j}\hat{j}} + m(r_{cg,\hat{i}}^2 + r_{cg,\hat{k}}^2) & -I_{\hat{j}\hat{k}} - m(r_{cg,\hat{j}}r_{cg,\hat{k}}) \\ -I_{\hat{k}\hat{i}} - m(r_{cg,\hat{k}}r_{cg,\hat{i}}) & -I_{\hat{k}\hat{j}} - m(r_{cg,\hat{j}}r_{cg,\hat{k}}) & I_{\hat{k}\hat{k}} + m(r_{cg,\hat{i}}^2 + r_{cg,\hat{j}}^2) \end{bmatrix} \quad (4.13)$$

$$I_{pl}^* = \begin{bmatrix} I_{pl,\hat{i}\hat{i}} + m_{pl}(r_{pl,\hat{j}}^{*2} + r_{pl,\hat{k}}^{*2}) & -I_{pl,\hat{i}\hat{j}} - m_{pl}(r_{pl,\hat{i}}^*r_{pl,\hat{j}}^*) & -I_{pl,\hat{i}\hat{k}} - m_{pl}(r_{pl,\hat{i}}^*r_{pl,\hat{k}}^*) \\ -I_{pl,\hat{j}\hat{i}} - m_{pl}(r_{pl,\hat{i}}^*r_{pl,\hat{j}}^*) & I_{pl,\hat{j}\hat{j}} + m_{pl}(r_{pl,\hat{i}}^{*2} + r_{pl,\hat{k}}^{*2}) & -I_{pl,\hat{j}\hat{k}} - m_{pl}(r_{pl,\hat{j}}^*r_{pl,\hat{k}}^*) \\ -I_{pl,\hat{k}\hat{i}} - m_{pl}(r_{pl,\hat{k}}^*r_{pl,\hat{i}}^*) & -I_{pl,\hat{k}\hat{j}} - m_{pl}(r_{pl,\hat{k}}^*r_{pl,\hat{j}}^*) & I_{pl,\hat{k}\hat{k}} + m_{pl}(r_{pl,\hat{i}}^{*2} + r_{pl,\hat{j}}^{*2}) \end{bmatrix}. \quad (4.14)$$

where the subscripts \hat{i} , \hat{j} and \hat{k} refer to the Cartesian components of a given vector or matrix in frame B . The resulting multirotor parameters after the payload configuration are

$$I_{cg} \leftarrow I_{cg}^* + I_{pl}^* \quad (4.15)$$

$$m \leftarrow m + m_{pl} \quad (4.16)$$

$$\mathbf{r}_i \leftarrow \mathbf{r}_i + \mathbf{r}_{cg} \quad (4.17)$$

4.1.8 Disturbance

According to the stated on Section 4.1, disturbance is considered as a linear 3D force included in the multirotor dynamic model. Let $\dot{\mathbf{v}}_A$ be the body acceleration on frame A , according to Equation (2.10), and let \mathbf{f}_{dist} be an arbitrary disturbance force. The body acceleration becomes, therefore,

$$\dot{\mathbf{v}}_A \leftarrow \dot{\mathbf{v}}_A + \frac{\mathbf{f}_{dist}}{m}. \quad (4.18)$$

The @multicopter class allows the simulation of any disturbance force described by a function of time. For the purpose of this work, the disturbance function of Equation (4.19) was considered, where A_d is the disturbance maximum value and t_1 and t_2 are the disturbance peaks occurrence time.

$$\mathbf{f}_{dist}(t) = [0, 1, 0]^T A_d (e^{-2(t-t_1)^2} + e^{-2(t-t_2)^2}) \quad (4.19)$$

4.1.9 Pseudo-Codes

Simulation sequence is summarized by Algorithm 2, which calls all methods presented in this work. This function is a method contained in the @multicontrol class. Appendix D presents the pseudo-codes for the main algorithms explained before and used in Algorithm 2. All details regarding variables handling, logging, and other programming specificities are not shown here for brevity.

Algorithm 2 Method "run" of class @multicontrol to simulate controllers actuating on multirotor dynamics

```

1: function RUN@MULTICONTROL(endTime, endError)
2:    $\delta t \leftarrow$  controller loop time step
3:   time  $\leftarrow \delta t$ 
4:   error  $\leftarrow 0$ 
5:   currState  $\leftarrow$  initial states read from previous user interface
6:   while time  $\leq$  endTime and error  $\leq$  endError do
7:     desState, desYaw  $\leftarrow$  DESIREDTRAJECTORY(time)
8:     diagnosis  $\leftarrow$  FDD(time)
9:     posCtrl  $\leftarrow$  POSITIONCONTROL(desState, currState)
10:    desAtt  $\leftarrow$  ATTITUDEREFERENCE(posCtrl, desYaw, diagnosis)
11:    attCtrl  $\leftarrow$  ATTITUDECONTROL(desAtt, posCtrl, diagnosis)
12:    actInput  $\leftarrow$  CONTROLALLOCATION(posCtrl, attCtrl, diagnosis)
13:    EVALCOMMANDS(time)       $\triangleright$  generates a simulated fault when programmed
14:    inputTime  $\leftarrow$  time  $- (1 - controlDelay) * \delta t$ 
15:    currState  $\leftarrow$  RUN@MULTICOPTER(actInput, inputTime, time)
16:    error  $\leftarrow$  DISTANCE(position, desiredPosition)            $\triangleright$  euclidian norm
17:    if error  $\geq$  endError or error is not a number then
18:      | simulationSuccess  $\leftarrow 1 - \frac{|end\ time - time|}{end\ time}$ 
19:    else
20:      | simulationSuccess  $\leftarrow 1$ 
21:    end if
22:    time  $\leftarrow$  time  $+ \delta t$ 
23:  end while
24: end function

```

4.2 Control allocation comparison

Three control allocation structures were presented on Section 3.5, namely Pseudo-Inverse (Section 3.5.1), Redistributed Pseudo-Inverse (Section 3.5.2) and Null-space based control allocation (Section 3.5.3). In order to choose a control allocation method for next sections, a simple comparison was performed.

The exact same simulation was performed for each control allocator using the simulation structure of Algorithm 2. Note that the same algorithm used for the control allocation is used partially for the attitude reference algorithm, as suggested on Section 3.3 and clarified in Algorithm 2. The simulation consisted of Model 2 performing a trajectory of 15 seconds in the form of a Lemniscate of Gerono (LAWRENCE, 1972) of length, width and height of 7 m, 4 m and 4 m, respectively with motors 3 and 1 faults at 0 and 7.5 seconds, respectively. The attitude reference was the \hat{z} axis with a rotating yaw. Each simulation used the exact same parameters for the PID trajectory controller and a PID attitude controller, in the sense that any divergence in the multirotor response was going to be caused by differences between the control allocators. Only the passive control allocators are compared here, resulting in three different simulations. The controllers parameters are

not important here.

4.3 Null-space based control allocation

To further investigate the potential of the control allocation algorithm of Section 3.5.3, three simulation cases were performed using the same simulation structure of Algorithm 2: a Nonsymmetric ennearotor, an octarotor with four outside-directed propellers and an hexarotor with two large and four small propellers; namely Case 1, Case 2 and Case 3, respectively. Tables 9, 10 and 11 present the multirotor parameters used for Cases 1, 2 and 3, respectively.

The trajectory used for all cases was the same as in Section 4.2. In all cases, the attitude reference q_ψ used was the \hat{z} axis with a fixed yaw in 0° or rotating 360° . It is important to note that the exact same code was used for all simulations, changing only the multirotor parameters to be simulated. The PID coefficients were not optimized, giving importance only to the attitude planning and control allocation features.

Figure 13 presents the three simulated architectures, where red arrows indicate negative rotation and gray arrows indicate lift direction, when applicable.

Table 9: Multirotor parameters for Case 1

Parameter	Value
\mathbf{r}_1	$[0.1125, 0, 0]^T m$
\mathbf{r}_2	$[0.0348, 0.1070, 0]^T m$
\mathbf{r}_3	$[-0.0910, 0.0661, 0]^T m$
\mathbf{r}_4	$[-0.0910, -0.0661, 0]^T m$
\mathbf{r}_5	$[0.0348, -0.1070, 0]^T m$
\mathbf{r}_6	$[0.3375, 0, 0]^T m$
\mathbf{r}_7	$[-0.1512, 0.3017, 0]^T m$
\mathbf{r}_8	$[-0.3362, 0.0298, 0]^T m$
\mathbf{r}_9	$[-0.2020, -0.2704, 0]^T m$
I_{cg}	$diag([4.856e^{-3}, 4.856e^{-3}, 8.801e^{-3}]) \text{ kg.m}^2$
μ_B	$diag([0.25, 0.25, 0.25]) \text{ kg.s}^{-1}$
$\mathbf{n}_1, \dots, \mathbf{n}_9$	$[0, 0, 1]^T$
m	0.468 kg
C_{L1}, \dots, C_{L9}	1.5^{-6} kg.m
C_{D1}, \dots, C_{D9}	1.0^{-7} kg.m^2
I_1, \dots, I_9	$6.357^{-5} \text{ kg.m}^2$
$[d_1, \dots, d_9]$	$[-1, 1, -1, 1, -1, 1, -1, 1, -1]^T$
$\omega_1^{max}, \dots, \omega_9^{max}$	1805 rad/s
$\omega_1^{min}, \dots, \omega_9^{min}$	0 rad/s

Table 10: Multirotor parameters for Case 2

Parameter	Value
\mathbf{r}_1	$[0.2250, 0, 0]^T m$
\mathbf{r}_2	$[0.1591, 0.1591, 0.1125]^T m$
\mathbf{r}_3	$[0, 0.2250, 0]^T m$
\mathbf{r}_4	$[-0.1591, 0.1591, 0.1125]^T m$
\mathbf{r}_5	$[-0.2250, 0, 0]^T m$
\mathbf{r}_6	$[-0.1591, -0.1591, 0.1125]^T m$
\mathbf{r}_7	$[0, -0.2250, 0]^T m$
\mathbf{r}_8	$[0.1591, -0.1591, 0.1125]^T m$
\mathbf{n}_1	$[1, 0, 0]^T$
\mathbf{n}_2	$[0, 0, 1]^T$
\mathbf{n}_3	$[0, 1, 0]^T$
\mathbf{n}_4	$[0, 0, 1]^T$
\mathbf{n}_5	$[-1, 0, 0]^T$
\mathbf{n}_6	$[0, 0, 1]^T$
\mathbf{n}_7	$[0, -1, 0]^T$
\mathbf{n}_8	$[0, 0, 1]^T$
I_{cg}	$diag([4.856e^{-3}, 4.856e^{-3}, 8.801e^{-3}]) \text{ kg.m}^2$
μ_B	$diag([0.25, 0.25, 0.25]) \text{ kg.s}^{-1}$
m	0.468 kg
C_{L1}, \dots, C_{L9}	2.980^{-6} kg.m
C_{D1}, \dots, C_{D9}	$1.140^{-7} \text{ kg.m}^2$
I_1, \dots, I_9	$3.357^{-5} \text{ kg.m}^2$
$[d_1, \dots, d_9]$	$[-1, -1, 1, 1, -1, -1, 1, 1]^T$
$\omega_1^{max}, \dots, \omega_8^{max}$	1805 rad/s
$\omega_1^{min}, \dots, \omega_8^{min}$	0 rad/s

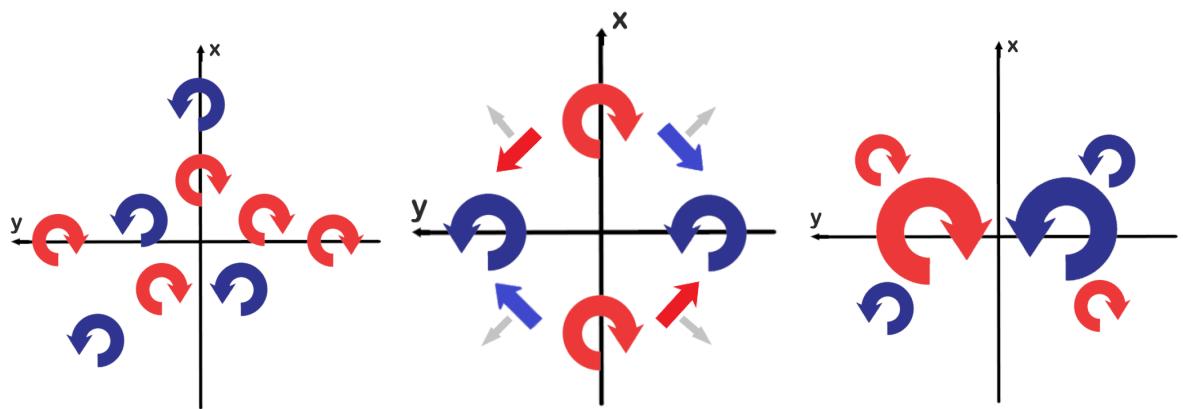


Figure 13: Rotor architectures for cases 1 to 3, from left to right, respectively.

Table 11: Multirotor parameters for Case 3

Parameter	Value
\mathbf{r}_1	$[2.41, 2.41, 0.6]^T m$
\mathbf{r}_2	$[-2.41, 2.41, 0.6]^T m$
\mathbf{r}_3	$[-2.41, -2.41, 0.6]^T m$
\mathbf{r}_4	$[2.41, -2.41, 0.6]^T m$
\mathbf{r}_5	$[1.55, 0, 0.6]^T m$
\mathbf{r}_6	$[-1.55, 0, 0.6]^T m$
I_{cg}	$diag([157, 61, 146]) \text{ kg.m}^2$
μ_B	$diag([0.25, 0.25, 0.25]) \text{ kg.s}^{-1}$
$\mathbf{n}_1, \dots, \mathbf{n}_6$	$[0, 0, 1]^T$
m	350 kg
C_{L1}, \dots, C_{L4}	$5.9664^{-04} \text{ kg.m}$
C_{L5}, C_{L6}	0.023 kg.m
C_{D1}, \dots, C_{D4}	$2.5102^{-05} \text{ kg.m}^2$
C_{D5}, C_{D6}	6.73^{-4} kg.m^2
$I_1 \dots I_4$	0.0234 kg.m^2
I_5, I_6	7.8626 kg.m^2
$[d_1, \dots, d_6]$	$[1, -1, 1, -1, 1, -1]^T$
$\omega_1^{max}, \dots, \omega_4^{max}$	600 rad/s
$\omega_5^{max}, \omega_6^{max}$	300 rad/s
$\omega_1^{min}, \dots, \omega_6^{min}$	0 rad/s

4.4 Controller optimization

Much attention is given to the development of diverse control architectures nowadays. Each architecture has a different tuning procedure. (BOYD; HAST; ÅSTRÖM, 2016) presents a tuning algorithm for MIMO PID structures while (MERHEB; BATEMAN; NOURA, 2015) uses the controller convergence assumptions along with evolutionary algorithms to choose the controller parameters. (SCHWAGER; ANNASWAMY; LAVRETSKY, 2005), (CERRI; TERRA, 2017), (CERRI, 2013) and (BORTOLIN; TERRA, 2015) do not provide any insights when it comes to tuning. In all cases, either many assumptions have to be made or only the expertise of the user is available for tuning the system. Considering the comparative analysis proposed in this work, the tuning procedure for all controllers here compared should be exempt of subjectivity, independent of the control architecture and focused on the same comparison metric for all. With this in mind, a genetic algorithm was chosen as the optimization method. The intention here was not to develop the topic of genetic algorithms but rather to use it as a tool. In this sense, the MATLAB toolkit for genetic algorithm (MATHWORKS, 2019a) was used.

Table 12: Chromosome composition

PIDD parameters (Equation (3.4))	Attitude controller parameters	α gains (Equation (4.1))
----------------------------------	--------------------------------	---------------------------------

Table 13: Genetic algorithm parameters

Parameter	Value
Population size	3000
Maximum number of generations	150
Crossover fraction	0.5
Mutation rate	0.05
Stall generation limit	25
Simulations per generation	6000

4.4.1 GA parameters

Table 13 presents the optimization parameters used during this work. For each controller, an individual chromosome was considered to be a single vector of type double of variables flattened in the format presented in Table 12. It is important to notice that every configurable parameter for an attitude controller was considered to be a parameter subject to optimization and, therefore, part of the chromosome of Table 12. This is particularly interesting in the case of the Robust controllers of Sections 3.4.4 and 3.4.5, where parameters that model the uncertainties are also treated like controller parameters. In this sense, this work expect these parameters to be optimized based on simulated scenarios of disturbance and variations.

Selection was made by the default uniform stochastic function, randomly choosing individuals for reproduction in a scale with scaled scores based on each individual position in the fitness rank. For reproduction, the default function of Scattered crossover was used, where a random binary vector mix the selected parents chromosomes to generate a child. The crossover fraction indicates the percentage of the next generation's population that will be generated by reproduction. At each generation, mutation was introduced with the uniform mutation function, where each individual gets a fraction of its genes changed to a random number inside an allowable range given a probability rate of being mutated. More details can be obtained at ([MATHWORKS, 2019b](#)).

4.4.2 Fitness function

Three metrics were considered in this work to measure how good a controller is for the optimization phase: Its ability to stabilize the aircraft, its ability to follow a commanded trajectory and the power it consumes to execute this trajectory. The first

metric consisted of how much of the simulation the controller was able to perform without diverging. The second metric consisted of the RMS error for the position and attitude, taken at each time step between the commanded trajectory and attitude and the actual trajectory and attitude for the entire trajectory. The third metric consisted of the RMS power for the trajectory.

At each generation, one individual was subject to two simulations. Both simulations consisted of the trajectory described in Section 4.2, rotating 360° around \hat{k} , and with four subsequent rotor failures at times

$$t_i = \frac{\delta t_{traj}(1 + n + i)}{1 + n}, \quad (4.20)$$

where δt_{traj} is the trajectory duration, n is the number of failures (four in this case), and i is the faulty rotor. The difference between simulations 1 and 2 is that one has no payload attached while the second considers a spherical payload at the position

$$\mathbf{r}_{pl} = \begin{bmatrix} 0, & 0, & -0.3 \sum_{i=1}^N \sqrt{r_{pl\hat{i}}^2 + r_{pl\hat{j}}^2 + r_{pl\hat{k}}^2} \end{bmatrix}, \quad (4.21)$$

with inertia tensor

$$I_{pl} = 1_{3x3} \frac{2m_{pl} \|\mathbf{r}_{pl}\|}{5}. \quad (4.22)$$

The fitness function for one individual at each generation is given by

$$fit = fit_1 + fit_2, \quad (4.23)$$

where fit_1 and fit_2 are the result of Equation (4.24) for each simulation. Note that the larger fit , the worse the controller according to the premisses above.

$$fit_{sim} = (1 - p_{sim}) + e_{p,RMS} + e_{att,RMS} + \frac{P_{RMS}}{\iota}, \quad (4.24)$$

where p_{sim} is the simulation success time percentage as presented in Algorithm 2,

$$e_{p,RMS} = \sqrt{\frac{1}{M} \sum_{k=0}^M \|\mathbf{p}^d(k\delta t_{sim}) - \mathbf{P}_A(k\delta t_{sim})\|^2}, \text{ for } M\delta t_{sim} = \delta t_{traj}, \quad (4.25)$$

is the position error,

$$e_{att,RMS} = \sqrt{\frac{1}{M} \sum_{k=0}^M \|\boldsymbol{\Upsilon}(q_{ref}(k\delta t_{sim})) - \boldsymbol{\Upsilon}(q(k\delta t_{sim}))\|^2}, \quad (4.26)$$

is the attitude error, P_{RMS} is the RMS power and ι is an arbitrary scaling factor. $\boldsymbol{\Upsilon}(q)$ represents the transformation from quaternion to Euler's *roll*, *pitch* and *yaw*, according to

$$\boldsymbol{\Upsilon}(q) = \begin{bmatrix} \text{atan2}(2q_2q_3 + 2q_0q_1, q_3^2 - q_2^2 - q_1^2 + q_0^2) \\ -\text{asin}(2q_1q_3 - 2q_0q_2) \\ \text{atan2}(2q_1q_2 + 2q_0q_3, q_1^2 + q_0^2 - q_3^2 - q_2^2) \end{bmatrix}. \quad (4.27)$$

Table 14: Fitness function simulation parameters

Parameter	Model 1	Model 2
δt_{traj} (s)	15.0	
m_{pl} for simulation 1 (kg)	0.0	
m_{pl} for simulation 2 (kg)	0.4	3.0
Disturbance A_d (N)	5.0	10.0
C_{L1}, \dots, C_{L8} (Ns ²)	$1 \cdot 10^{-6}$	$86.98 \cdot 10^{-6}$
C_{D1}, \dots, C_{D8} (Nm s ²)	$0.3 \cdot 10^{-6}$	$8.89 \cdot 10^{-7}$
FDD hit rate	0.95	
FDD delay (s)	0.5	
Solver	Euler	
ι	500	3000

The adaptive controllers were found to need a smaller control step period δt to converge. Therefore, for the adaptive controllers, $\delta t = 0.020$ s while for all other controllers $\delta t = 0.050$ s. Table 14 summarizes the fitness function parameters for controller optimization simulations following notation previously defined.

(BISSEL; CHAPMAN, 1992) states that the RMS value of a signal is the sum of its squared mean and its variance, such that

$$e_{p,RMS} = E^2(e_p) + V(e_p), \quad (4.28)$$

$$e_{att,RMS} = E^2(e_{att}) + V(e_{att}), \text{ and} \quad (4.29)$$

$$P_{RMS} = E^2(P) + V(P). \quad (4.30)$$

In this sense, the choice for the RMS of the errors and power consumption has the power of optimizing controllers to reduce both steady-state errors and oscillations at the same time.

4.4.3 Initial population

The initial population was created using one individual chosen based on manual simulations and the rest of individuals randomly created using a uniform distribution over the allowable values. The first individual was chosen in a manner that each controller had at least one individual that was able to stabilize the multirotor at least without considering any failures or parameter variations, reducing the GA's efforts in a search for a valid solution.

4.4.4 Optimization cases

Diverse controller structures were optimized or "tuned" for further comparison. Considering this, Tables 15 and 16 summarize each optimization case and names them in groups. Cases differ in relation to model, controller, control allocation method used in

Table 15: Optimization cases for passive controllers

Case	Model	Attitude controller	Attitude planning	Control allocation
1.1.1	Model 1	PID (Section 3.4.1)	Passive NSCA ¹	Passive NSCA
1.1.2.1	Model 1	SOSMC (Section 3.4.2.1)	Passive NSCA	Passive NSCA
1.1.2.2	Model 1	SOSMC with PIDD (Section 3.4.2.3)	Passive NSCA	Passive NSCA
1.1.2.3	Model 1	SOSMC Direct (Section 3.4.2.5)	Passive NSCA	None
1.1.3.1	Model 1	R-LQR (Section 3.4.4.1)	Passive NSCA	Passive NSCA
1.1.3.2	Model 1	R-LQR with rotor failures (Section 3.4.4.3)	Passive NSCA	Passive NSCA
1.1.3.3	Model 1	R-LQR with rotor failures and PIDD (Section 3.4.4.5)	Passive NSCA	Passive NSCA
1.1.4	Model 1	Mode-independent R-LQR for DMJLSU	Passive NSCA	Passive NSCA
1.1.5.1	Model 1	Adaptive (Section 3.4.3)	Passive NSCA	Passive NSCA
1.1.5.2	Model 1	Adaptive with PIDD (Section 3.4.3.2)	Passive NSCA	Passive NSCA
1.1.5.3	Model 1	Adaptive Direct (Section 3.4.3.3)	Passive NSCA	None
1.2.1	Model 2	PID (Section 3.4.1)	Passive NSCA	Passive NSCA
1.2.2.1	Model 2	SOSMC (Section 3.4.2.1)	Passive NSCA	Passive NSCA
1.2.2.2	Model 2	SOSMC with PIDD (Section 3.4.2.3)	Passive NSCA	Passive NSCA
1.2.2.3	Model 2	SOSMC Direct (Section 3.4.2.5)	Passive NSCA	None
1.2.3.1	Model 2	R-LQR (Section 3.4.4.1)	Passive NSCA	Passive NSCA
1.2.3.2	Model 2	R-LQR with rotor failures (Section 3.4.4.3)	Passive NSCA	Passive NSCA
1.2.3.3	Model 2	R-LQR with rotor failures and PIDD (Section 3.4.4.5)	Passive NSCA	Passive NSCA
1.2.4	Model 2	Mode-independent R-LQR for DMJLSU	Passive NSCA	Passive NSCA
1.2.5.1	Model 2	Adaptive (Section 3.4.3)	Passive NSCA	Passive NSCA
1.2.5.2	Model 2	Adaptive with PIDD (Section 3.4.3.2)	Passive NSCA	Passive NSCA
1.2.5.3	Model 2	Adaptive Direct (Section 3.4.3.3)	Passive NSCA	None

Table 16: Optimization cases for active controllers

Case	Model	Attitude controller	Attitude planning	Control allocation
2.1.1	Model 1	PID (Section 3.4.1)	Active NSCA	Active NSCA
2.1.2.1	Model 1	SOSMC (Section 3.4.2.2)	Active NSCA	Active NSCA
2.1.2.2	Model 1	SOSMC with PIDD (Section 3.4.2.4)	Active NSCA	Active NSCA
2.1.2.3	Model 1	SOSMC Direct (Section 3.4.2.5)	Active NSCA	None
2.1.3.1	Model 1	R-LQR (Section 3.4.4.1)	Active NSCA	Active NSCA
2.1.3.2	Model 1	R-LQR with rotor failures (Section 3.4.4.3)	Active NSCA	Active NSCA
2.1.3.3	Model 1	R-LQR with rotor failures and PIDD (Section 3.4.4.5)	Active NSCA	Active NSCA
2.1.4	Model 1	Mode-dependent R-LQR for DMJLSU (Section 3.4.5.1)	Active NSCA	Active NSCA
2.2.1	Model 2	PID (Section 3.4.1)	Active NSCA	Active NSCA
2.2.2.1	Model 2	SOSMC (Section 3.4.2.2)	Active NSCA	Active NSCA
2.2.2.2	Model 2	SOSMC with PIDD (Section 3.4.2.4)	Active NSCA	Active NSCA
2.2.2.3	Model 2	SOSMC Direct (Section 3.4.2.5)	Active NSCA	None
2.2.3.1	Model 2	R-LQR (Section 3.4.4.1)	Active NSCA	Active NSCA
2.2.3.2	Model 2	R-LQR with rotor failures (Section 3.4.4.3)	Active NSCA	Active NSCA
2.2.3.3	Model 2	R-LQR with rotor failures and PIDD (Section 3.4.4.5)	Active NSCA	Active NSCA
2.2.4	Model 2	Mode-dependent R-LQR for DMJLSU (Section 3.4.5.1)	Active NSCA	Active NSCA

attitude planning as suggested on Equation (3.9), and control allocation method. The final fitness values obtained in this optimization phase may suggest each controller's robustness level. The robustness evaluation, however, will consider diverse scenarios and metrics, as will be presented on Section 4.5.

¹ NSCA stands for Null-Space based Control Allocation

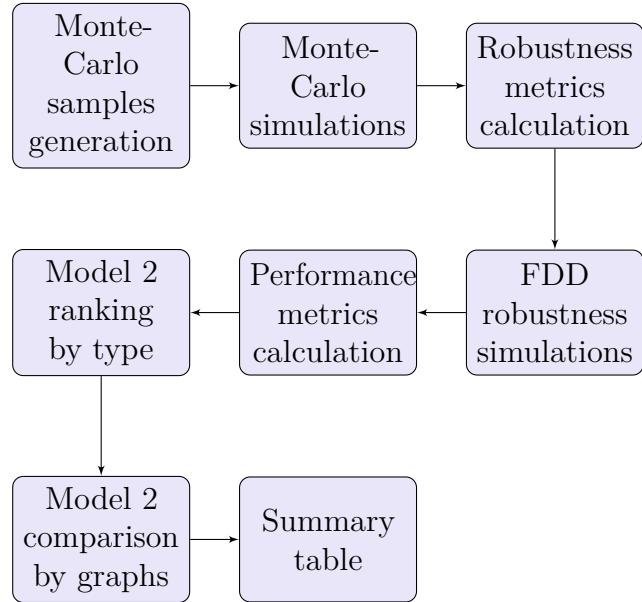


Figure 14: Comparative study flow

4.5 Controller evaluation

After optimizing each controller case using a non-subjective method presented on Section 4.4, the main goal of this work is to provide a comparative robustness analysis between each of them, as introduced on Section 1.4, and to analyse which are the main drawbacks and advantages of each control architecture.

The comparative study follows the flow presented on Figure 14. Simulation results were separated and analysed in chunks and will be presented and discussed as suggested by Figure 15. Model 1 cases were used to showcase, in contrast to Model 2 cases, differences in responses when choosing different models. Model 2 cases were then ranked so as to obtain a final selection of controllers each from a different architecture. These selected controllers were used to compare architectures based on robustness and performance within the groups of passive and active controllers. Later, passive and active architectures could be compared showcasing advantages and drawbacks of having or not FDD schemes.

4.5.1 Variance-based robustness analysis

There are diverse objectives when using variance-based sensitivity analysis, as introduced on Section 1.4. (SALTELLI et al., 2004) define the *Factors Priorization (FP) Setting* in sensitivity analysis. This analysis setting has the objective of ranking the factors in a system "according to the amount of output variance that is removed when we learn the true value of a given input factor X_i ". Besides analysing how much a system is robust in relation to its unconditional variance, we propose here to use sensitivity analysis of the system outputs to the system inputs as a measure of robustness of the system to each specific parameter or disturbance. Therefore, instead of focusing on which factor one

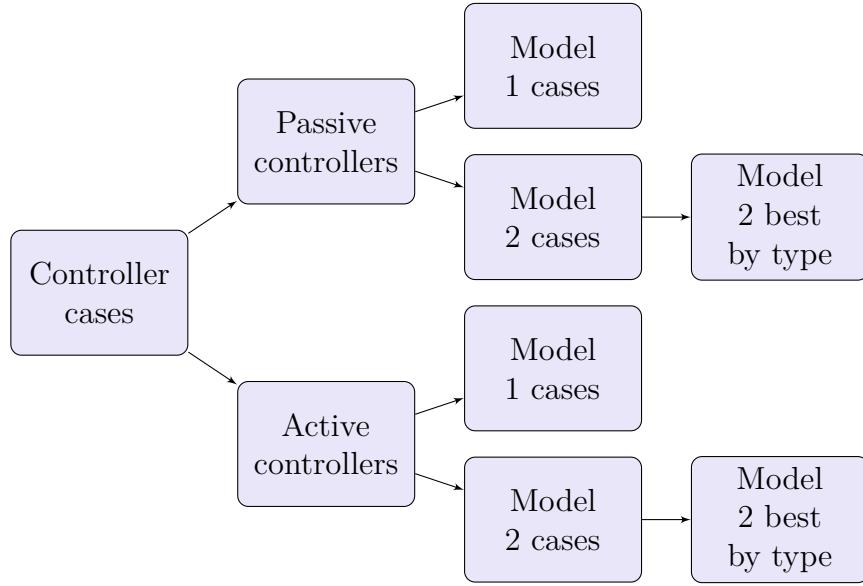


Figure 15: Clustering of comparison results

should "know the true value of", we will find which factor causes the system to variate more, and will be able to rank what parameters and disturbances are most likely to destabilize a controller under evaluation.

Considering this objective, let

$V(Y)$ be the unconditional variance of a system output Y ;

$V(Y|X_f) = V_f$ be the conditional variance of Y given a fixed value of the system parameter X_f . V_f is defined as a first-order effect;

$F = \{1, 2, \dots, k\} \subset \mathbb{N}$ represents the indices of parameters X_f while $F_s = \{f_1, \dots, f_s\} \subset F$ represents a closed selection of indices in F ;

$V(Y|X_{f_1}, \dots, X_{f_s}) = V_{f_1 f_2 \dots f_s}^c$ be the conditional variance of Y given fixed values of the system parameters X_{f_1}, \dots, X_{f_s} , where c indicates the closed set F_s ;

$V_{f_1 f_2} = V(E(Y|X_{f_1}, X_{f_2})) - V_{f_1} - V_{f_2}$ be a second-order effect, which is the variance caused solely by the interaction between parameters X_{f_1} and X_{f_2} ; and

$V_{f_1 f_2 f_3} = V(E(Y|X_{f_1}, X_{f_2}, X_{f_3})) - V_{f_1 f_2} - V_{f_1 f_3} - V_{f_2 f_3} - V_{f_1} - V_{f_2} - V_{f_3}$ be a third-order effect.

(SALTELLI et al., 2004) states that the contribution of an isolated parameter f to the system variance $V(Y)$ can be calculated by means of the *sensitivity index*

$$S_f = \frac{V_f}{V(Y)}. \quad (4.31)$$

Meanwhile, the *total effect index*

$$S_{Tf1} = S_{f1} + S_{f1f2} + \dots + S_{f1f2\dots fk} \quad (4.32)$$

gives the total contribution to the system variance of a parameter X_{f1} and its interactions with all other parameters X_{f2}, \dots, X_{fk} . In this sense, $S_{Tf} - S_{f1}$ gives the sum of variance contributions of only the interactions between X_{f1} and X_{f2}, \dots, X_{fk} .

One can conclude for a given controller, therefore, that the greater S_{Tf} , the less robust the controller is to the parameter f and its interactions to other parameters.

4.5.2 Variance and expected value estimation

Let

$V_{f1f2\dots fs}^c = \sum_{f \in F_s} V_f + \sum_{f \in F_s} \sum_{g > f, f, g \in F_s} V_{fg} + \dots + V_{f1f2\dots fs}$ be the decomposition representation of $V(Y|X_{f1}, \dots, X_{fs})$;

$F_s^c = F \setminus F_s$ be the complement of F_s in F ;

$V_{-f1f2\dots fs}^c$ represent the same decomposition as $V_{f1f2\dots fs}^c$ but on the subset F_s^c .

The set of parameters and disturbances are considered orthogonal in this work, in the sense that they are independent of each other. For non-orthogonal inputs, the calculation of sensitivity indices would call for the straightforward calculation of multidimensional integrals over the space of the input variables. (SALTELLI et al., 2004) presents an estimation method for \hat{S}_f , \hat{S}_{Tf} and \hat{V}_{-fg}^c (of order $k - 2$), for all k indices $f, g \in F$ at the cost of $W(k + 2)$ simulations. Superscript $\hat{\cdot}$ denotes the estimate of a parameter while W is the number of simulation samples, where each sample is composed of one fixed value for each parameter X_f .

Considering the number of parameters analysed, which will be presented next on Section 4.5.3, it is unfeasible to simulate all possible parameter variations. Therefore, a Monte-Carlo simulation is necessary where, simply stating, one can estimate probability distributions from a large pool of independent and identically distributed random samples (DOUCET; FREITAS; GORDON, 2001).

Consider, therefore, input sample matrices

$$M_1 = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_k^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_k^{(2)} \\ \vdots & \vdots & \dots & \vdots \\ x_1^{(W)} & x_2^{(W)} & \dots & x_k^{(W)} \end{bmatrix} \quad (4.33)$$

and

$$M_2 = \begin{bmatrix} x_1^{(1')} & x_2^{(1')} & \dots & x_k^{(1')} \\ x_1^{(2')} & x_2^{(2')} & \dots & x_k^{(2')} \\ \vdots & \vdots & \dots & \vdots \\ x_1^{(W')} & x_2^{(W')} & \dots & x_k^{(W')} \end{bmatrix} \quad (4.34)$$

where each row represents a sample for one controller simulation and each column is drawn from the marginal distribution of each parameter X_f . Consider a third matrix

$$N_f = \begin{bmatrix} x_1^{(1')} & x_2^{(1')} & \dots & x_f^{(1)} & \dots & x_k^{(1')} \\ x_1^{(2')} & x_2^{(2')} & \dots & x_f^{(2)} & \dots & x_k^{(2')} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ x_1^{(W')} & x_2^{(W')} & \dots & x_f^{(W)} & \dots & x_k^{(W')} \end{bmatrix} \quad (4.35)$$

made up of elements from M_2 where only factor X_f is taken from matrix M_1 . Accordingly, $N_{f1...fs}$ is obtained from M_2 by substituting factors X_{f1}, \dots, X_{fs} .

The *sensitivity index* S_f can be estimated by

$$\hat{S}_f = \frac{\hat{U}_f - \hat{E}^2(Y)}{\hat{V}(Y)}, \quad (4.36)$$

where

$$\hat{U}_f = \frac{1}{W-1} \sum_{r=1}^W f(x_1^{(r)}, x_2^{(r)}, \dots, x_k^{(r)}) f(x_1^{(r')}, x_2^{(r')}, \dots, x_{f-1}^{(r')}, x_f^{(r)}, x_{f+1}^{(r')}, \dots, x_k^{(r')}), \quad (4.37)$$

$$\hat{V}(Y) = \frac{1}{W-1} \sum_{r=1}^W f^2(x_1^{(r)}, x_2^{(r)}, \dots, x_k^{(r)}) - \hat{E}^2(Y) \quad (4.38)$$

and

$$\hat{E}^2(Y) = \frac{1}{W} \sum_{r=1}^W f(x_1^{(r)}, x_2^{(r)}, \dots, x_k^{(r)}) f(x_1^{(r')}, x_2^{(r')}, \dots, x_k^{(r')}). \quad (4.39)$$

$f(\cdot)$ here represents a scalar result from the controller evaluation from a specified row from either M_1 , M_2 or $N_{f1...fs}$.

The *total effect index* S_{Tf} can be estimated by

$$\hat{S}_{Tf} = 1 - \frac{\hat{U}_{-f} - \hat{E}^2(Y)}{\hat{V}(Y)}, \quad (4.40)$$

where

$$\hat{U}_{-f} = \frac{1}{W-1} \sum_{r=1}^W f(x_1^{(r)}, x_2^{(r)}, \dots, x_k^{(r)}) f(x_1^{(r)}, x_2^{(r)}, \dots, x_{f-1}^{(r)}, x_f^{(r)}, x_{f+1}^{(r)}, \dots, x_k^{(r)}), \quad (4.41)$$

which is equivalent to

$$\hat{U}_{-f} = \frac{1}{W-1} \sum_{r=1}^W f(x_1^{(r')}, x_2^{(r')}, \dots, x_k^{(r')}) f(x_1^{(r')}, x_2^{(r')}, \dots, x_{f-1}^{(r')}, x_f^{(r)}, x_{f+1}^{(r')}, \dots, x_k^{(r')}). \quad (4.42)$$

In this case,

$$\hat{E}^2(Y) = \left(\frac{1}{W} \sum_{r=1}^W f(x_1^{(r)}, x_2^{(r)}, \dots, x_k^{(r)}) \right)^2, \quad (4.43)$$

or equivalently,

$$\hat{E}^2(Y) = \left(\frac{1}{W} \sum_{r=1}^W f(x_1^{(r')}, x_2^{(r')}, \dots, x_k^{(r')}) \right)^2, \quad (4.44)$$

according to what is suggested by (SALTELLI et al., 2004).

Let $\mathbf{a}_{f1\dots fs}$ be the vector of evaluations of $f(\cdot)$ over each row of $N_{f1\dots fs}$ such that

$$\mathbf{a}_0 = f(M_2) \quad (4.45)$$

$$\mathbf{a}_{12\dots k} = f(M_1) \quad (4.46)$$

$$\mathbf{a}_{f1\dots fs} = f(N_{f1\dots fs}). \quad (4.47)$$

For the estimation of all \hat{S}_f and \hat{S}_{Tf} , it is necessary to have only the results of set $A_k = \{\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k, \mathbf{a}_{12\dots k}\}$, which comprise Equations (4.37), (4.38), (4.39), (4.41) and (4.43) in terms of scalar products $\mathbf{b} \cdot \mathbf{c}$ for $\mathbf{b}, \mathbf{c} \in A_k$.

4.5.3 Monte-Carlo Samples

Seven effects were considered of interest for robustness analysis in this work. They are either a combination of or isolated parameters presented before. These effects are

- Payload: represented by the payload mass as a percentage $0 \leq m_{pl\%} \leq 1$ of the multirotor mass, such that $m_{pl} = m_{pl\%}m$, \mathbf{r}_{pl} is fixed and I_{pl} is dependent on m_{pl} , according to Equations (4.21) and (4.22) respectively;
- Aircraft velocity and acceleration: represented by the time δt_{traj} required to complete the trajectory (Section 4.4.2);
- Disturbance: represented by an acceleration A_d/m ;
- Drag and Lift coefficients: represented by $c_{1/0} \in [0, 1]$. $c_{1/0}$ dictates whether simulation will consider drag and lift coefficients to be variable ($c_{1/0} = 1$) or not ($c_{1/0} = 0$), as presented on Section 4.1.4;
- Rotor failures: represented by a finite set $Z = \{\mathbf{z} = [\xi_1, \dots, \xi_N, \xi_{P1}, \dots, \xi_{PN}]\}$. Z is presented on Table 17;
- Control loop time: represented by δt ;
- Control delay percentage: as presented on Section 4.1.2 and defined here as $\Delta t \in \mathbb{R} \mid 0 < \Delta t < 1$.

Table 17: Rotor failures set Z

z	Motor and propeller efficiencies
z_1	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
z_2	[0.78, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
z_3	[0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
z_4	[0, 0.78, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
z_5	[0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
z_6	[0, 0, 0.78, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
z_7	[0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
z_8	[0, 0, 0, 0.78, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
z_9	[0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
z_{10}	[0, 0, 1, 1, 1, 1, 0, 0.78, 1, 1, 1, 1, 1, 1, 1]
z_{11}	[0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1]
z_{12}	[1, 1, 1, 1, 1, 1, 1, 0.1, 1, 1, 1, 1, 1, 1]
z_{13}	[1, 1, 1, 1, 1, 1, 1, 0.1, 0.1, 1, 1, 1, 1, 1]
z_{14}	[1, 1, 1, 1, 1, 1, 1, 0.1, 0.1, 0.1, 1, 1, 1, 1]
z_{15}	[1, 1, 1, 1, 1, 1, 1, 0.1, 0.1, 0.1, 0.1, 1, 1, 1]
z_{16}	[1, 1, 1, 1, 1, 1, 1, 0.1, 0.1, 1, 1, 1, 0.1, 0.1]

Table 18: Parameters X_f

Parameter	Probability Distribution
X_1	$m_{pl}\%$
X_2	$\mathcal{U}\{10, 20\}$
X_3	A_d/m
X_4	$\mathcal{U}_{\mathbb{N}}\{0, 1\}$
X_5	z_l l random variable with $\mathcal{U}_{\mathbb{N}}\{1, 16\}$
X_6	δt
X_7	Δt

Some parameters may seem to be non-orthogonal, like the payload inertia dependability on payload mass. However, what is being analysed here is the combined effects between some parameters. These effects vary independently between each other and can thus be considered orthogonal. Table 18 summarizes the evaluation parameters and their probability distribution. There were $W = 1500$ Monte-Carlo samples, which gives a total of 13500 Monte-Carlo simulations for each controller evaluation considering the set A_k . \mathcal{U} represents the continuous Uniform Probability Distribution while \mathcal{U}_d represents the discrete Uniform Probability Distribution in \mathbb{N} .

It is important to notice that matrices M_1 and M_2 were generated only once using Table 18. After that, all controller evaluations used the same sample matrices so as to guarantee comparability.

FDD hit rate and delay were not considered Monte-Carlo parameters as explained

on Section 4.5.6. The hit rate and delay for active controller cases, however, were considered to be 99% and 0.2s, respectively.

4.5.4 Robustness metrics

Robustness metrics proposed here have the purpose of analysing stability over the range of parameter variations and of analysing what parameters most compromise stability for each controller. The metrics are presented below.

Success rate S_r : For each simulation sample, a percentage p_{sim} of the total simulation that was executed before reaching a maximum position error of 5 m is calculated. The success rate is the percentage of simulations, from the total of $W(k + 2)$, that present $p_{sim} = 1$. This quantifies how much a given controller can stabilize the system through the duration of the entire mission over the range of parameter variations. Therefore, the greater S_r , the more robust the controller in general.

Mean simulation success $E(p_{sim})$: Indicates the average of mission, or simulation trajectory completeness percentage for all simulation cases.

Simulation success unconditional variance $\hat{V}(p_{sim})$: Is the unconditional variance of Equation (4.38) for Y equals to p_{sim} . The lower $\hat{V}(p_{sim})$, the more robust the controller is regarding system stability.

Simulation success total effect index $\hat{S}_{Tf,p_{sim}}$: Is the total effect index for each parameter, according to Equation (4.32), for Y equals to p_{sim} . The greater $\hat{S}_{Tf,p_{sim}}$, the less robust the controller is in relation to parameter X_f and its interactions regarding system stability.

Simulation success sensitivity index $\hat{S}_{f,p_{sim}}$: Is the sensitivity index for each parameter, according to Equation (4.31), for Y equals to p_{sim} . The greater $\hat{S}_{f,p_{sim}}$, the less robust the controller is in relation to parameter X_f alone regarding system stability.

Position error unconditional variance $\hat{V}(e_{p,RMS})$: Is the unconditional variance of Equation (4.38) for Y equals to $e_{p,RMS}$. The lower $\hat{V}(e_{p,RMS})$, the more robust the controller is regarding position control.

Position error total effect index $\hat{S}_{Tf,pos}$: Is the total effect index for each parameter, according to Equation (4.32), for Y equals to $e_{p,RMS}$. The greater $\hat{S}_{Tf,pos}$, the less robust the controller is in relation to parameter X_f and its interactions regarding position control.

Position error sensitivity index $\hat{S}_{f,pos}$: Is the sensitivity index for each parameter, according to Equation (4.31), for Y equals to $e_{p,RMS}$. The greater $\hat{S}_{f,pos}$, the less robust the controller is in relation to parameter X_f alone regarding position control.

RMS power unconditional variance $\hat{V}(P_{RMS})$: Is the unconditional variance of Equation (4.38) for Y equals to P_{RMS} . The lower $\hat{V}(P_{RMS})$, the more robust the controller is regarding power consumption.

RMS power total effect index $\hat{S}_{Tf,pow}$: Is the total effect index for each parameter, according to Equation (4.32), for Y equals to P_{RMS} . The greater $\hat{S}_{Tf,pow}$, the less robust the controller is in relation to parameter X_f and its interactions regarding power consumption.

RMS power sensitivity index $\hat{S}_{f,pow}$: Is the sensitivity index for each parameter, according to Equation (4.31), for Y equals to P_{RMS} . The greater $\hat{S}_{f,pow}$, the less robust the controller is in relation to parameter X_f alone regarding power consumption.

4.5.5 Performance metrics

Performance metrics proposed here have the purpose of analysing how good a controller is after guaranteeing stability to the system. This implies that these metrics are calculated and then averaged only over the range of simulations that present $p_{sim} = 1$.

Mean absolute position error $E(e_p)$: The less $E(e_p)$, the better the controller is regarding position tracking.

Mean local variance of position error $E(V(e_p))$: The less $E(V(e_p))$, the less position oscillations the controller generates.

Mean absolute attitude error $E(e_{att})$: The less $E(e_{att})$, the better the controller is regarding attitude tracking.

Mean local variance of attitude error $E(V(e_{att}))$: The less $E(V(e_{att}))$, the less attitude oscillations the controller generates.

Mean absolute power consumption $E(P)$: The less $E(P)$, the better the controller is regarding power consumption.

Mean local variance of power consumption $E(V(P))$: The less $E(V(P))$, the less chattering the controller generates, considering that the greater part of rotor consumption variance comes from rotor acceleration.

Computational cost J_c : This is the mean time taken to compute one attitude reference, attitude control and control allocation iteration. This does not include the position control which is the same for all controller cases. The method used was Matlab's "tic" and "toc" functions.

Position errors and power consumption are defined in Section 4.4.2. By "local" we refer to the metric for one sample simulation, so as to not get in confusion with the variances caused by variation of parameters X_f as presented before.

4.5.6 FDD robustness evaluation

FDD hit rate and delay were not considered as Monte-Carlo parameters. FDD simulation is a random process for itself and therefore, for a better representation of the system, a larger Monte-Carlo sample set would be necessary. That would take too much time and computational effort not available.

Considering this, FDD influence on active controllers was evaluated individually. After finished Monte-Carlo simulations, the best Monte-Carlo sample from simulations for each active controller case was chosen based on the fitness function from Section 4.4.2. This sample was then used to simulate the same controller case varying the FDD hit rate from 100% to 88% and the FDD delay from 0 to 3 s, separately. For each case, at least ten simulations were performed in order to consider the random behaviour of the FDD algorithm.

5 RESULTS

This Chapter presents all results for this work. Section 5.1 compares all three control allocation methods proposed, evidencing the best one. Section 5.2 deepens the understanding of the Null-Space based control allocation. Section 5.3 presents optimization evolution and the parameters results. Section 5.4 presents the main results of this work. It first introduces the Monte-Carlo samples used for controller evaluations and then presents tables and graphs with all the proposed evaluation metrics.

5.1 Control alocation comparison

First, consider Model 2 with the modification of rotor orientations proposed in Table 19. The trajectories for this case, considering the three passive algorithms of Section 3.5 are presented on Figure 16 while their respective rotor 1 speeds are presented on Figure 17. Note that in this case, the control allocation algorithms performances are exactly the same for the trajectory and very similar for rotor 1 speed even in the presence of actuator failures.

Consider now the actual rotor orientations of Model 2 as presented on Table 8. The

Table 19: Canonical rotor orientations

Parameter	Value
$\mathbf{n}_1, \dots, \mathbf{n}_6$	$[0, 0, 1]^T$

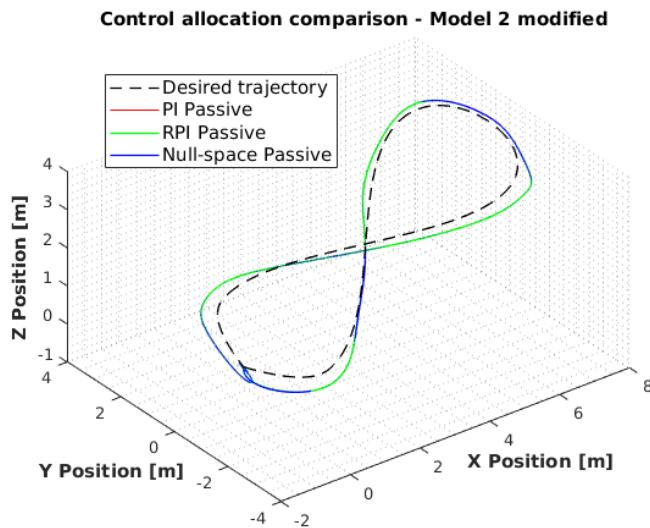


Figure 16: Control allocation trajectory comparison for Model 2 modified.

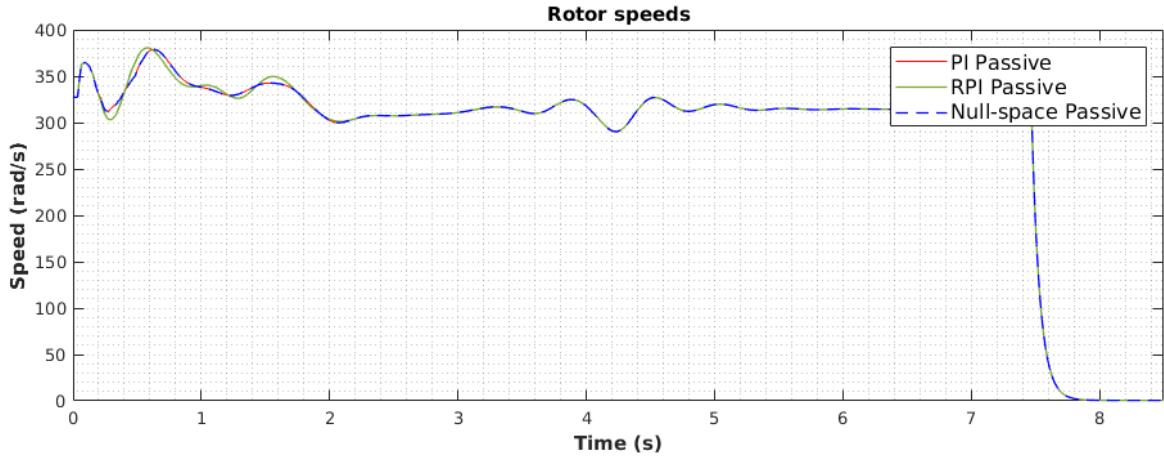


Figure 17: Control allocation rotor speeds comparison for Model 2 modified.

simulation trajectories are presented on Figure 18 and rotor 1 speeds are presented on Figure 19. Note that the Null-space based algorithm is the only one capable of stabilizing the multirotor.

The lack of robustness of the Pseudo-Inverse and Redistributed Pseudo-Inverse for this case is due mainly to the fact that the force matrix M_f for Model 2 has rank 3 instead of rank 1 like the force matrix for a model with rotor orientations of Table 19. In the latter case, \mathbf{T}_C^d will always be projected on \hat{k} or any arbitrary axis to which all rotors may point to, keeping $\boldsymbol{\Omega}$ within operational speeds. If $\text{rank}(M_f) > 1$, however, some \mathbf{T}_C^d may require negative components of $\boldsymbol{\Omega}$, which are not allowed. These negative values will be saturated by Equation (3.10), causing the behaviour of Figure 19.

The Pseudo-Inverse guarantees the closest solution $\boldsymbol{\Omega}$ for \mathbf{T}_C^d (BARATA; HUSSEIN, 2012), which is a problem for any multirotor which $\text{rank}(M_f) > 1$. The cost function of Equation (3.130), however, finds the closest solution to \mathbf{T}_C^d but at the same time to $\boldsymbol{\Omega}_{1/2}$, according to weights W_m and W_a . Since $\boldsymbol{\Omega}_{1/2}$ is always positive, the solution $\boldsymbol{\Omega}$ will not saturate. As W_a increases, the $\boldsymbol{\Omega}$ will get closer to fulfill \mathbf{T}_C^d but guaranteeing positiveness. This kind of solution works for arbitrary rotor orientations and allow also the change from sub-actuated to fully-actuated behaviour as presented in Section 5.2.2.

5.2 Null-space based control allocation

This Section will present the simulation results from Section 4.3. Sections 5.2.1, 5.2.2 and 5.2.3 present Cases 1, 2 and 3 respectively.

5.2.1 Case 1: Nonsymmetric ennearotor

This case highlights the algorithm's potential to deal with an arbitrary number of rotors positioned arbitrarily. Here, the multirotor tries to keep the yaw fixed in zero

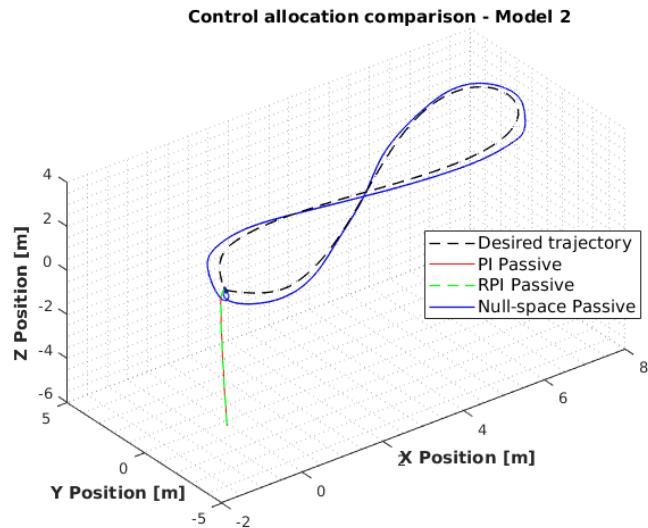


Figure 18: Control allocation trajectory comparison for Model 2.

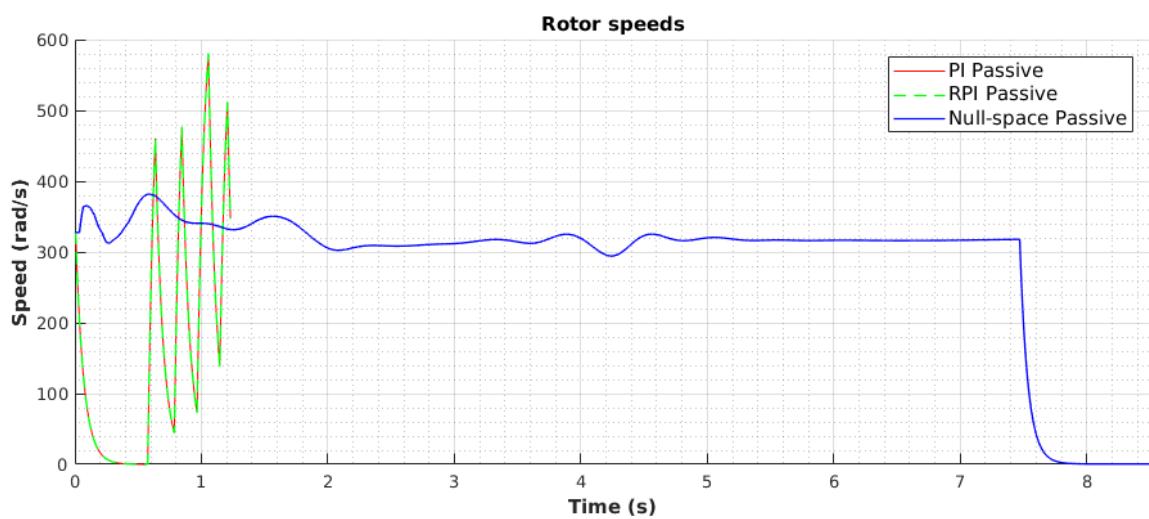


Figure 19: Control allocation rotor speeds comparison for Model 2.

degrees. Figure 20 presents the ennearotor 3D trajectory and Figure 21 presents rotor speeds allocation for the proposed architecture.

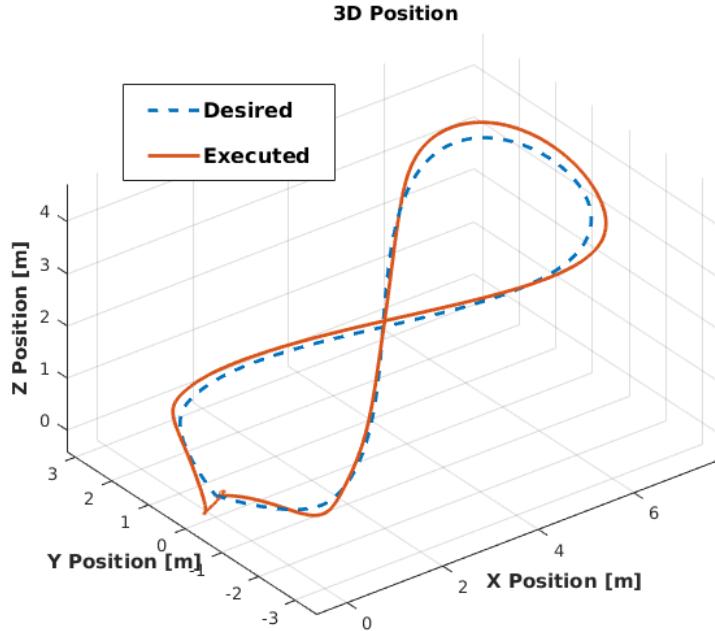


Figure 20: Trajectory for Case 1. The same applies to Cases 2 and 3.

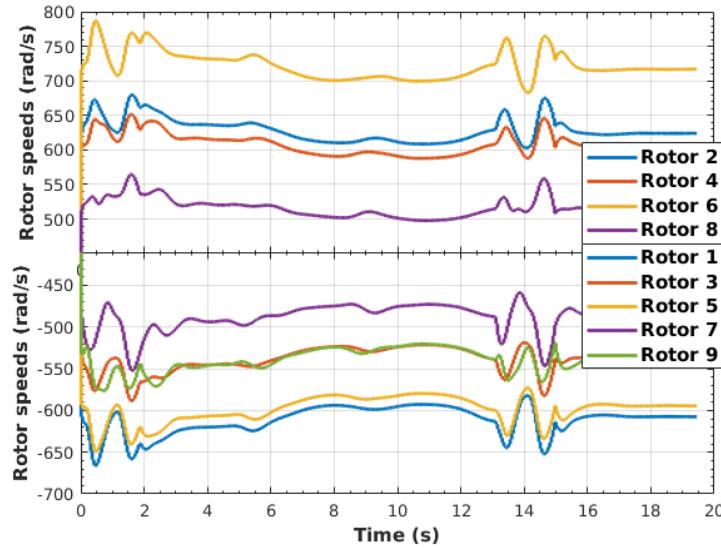


Figure 21: Rotor speeds allocation for nonsymmetric ennearotor.

5.2.2 Case 2: Fully-actuated octarotor

The goal in this case is to explore the effect of weights W_m and W_a on multirotor attitude. This multirotor is composed of a quadrotor like structure on top and four rotors generating thrust to the sides of the multirotor. Here, the multirotor rotates 360° once around its \hat{k} ax during the entire trajectory.

5.2.2.1 $W_m = 1$ and $W_a = 0$

By favouring maneuverability in this case, the multirotor behaves almost like a regular quadrotor. This happens because, since rotor speeds tend to be close to half the allowable range, the thrust generated by the four lower rotors cancel each other. In this sense, the multirotor has to tilt to move in space. This can be observed on the blue curves of Figure 24 where roll and pitch have large peaks. Figure 22 presents the control allocation for this simulation case.

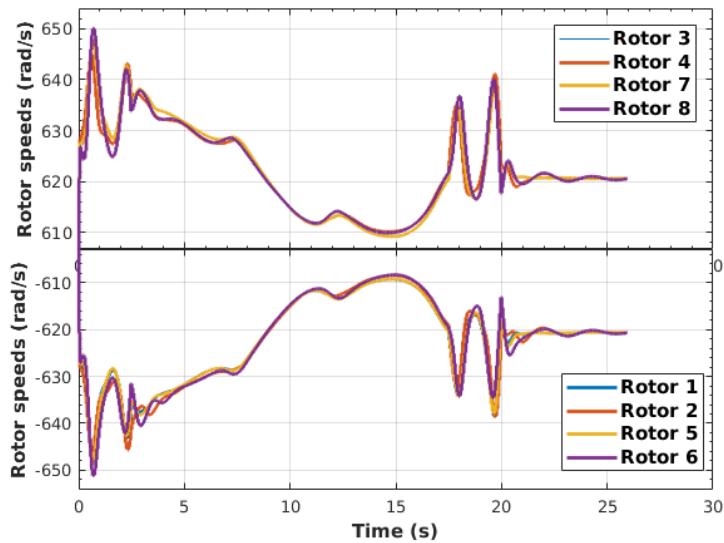


Figure 22: Rotor speeds allocation for $W_a = 0$ in Case 2.

5.2.2.2 $W_m = 1$ and $W_a = 1.2 \cdot 10^{11}$

When $W_a = 1.2 \cdot 10^{11}$, the multirotor tries to generate \mathbf{T}_B^d and to keep the attitude reference of \hat{k} aligned with \hat{z} . In this case, it behaves like a fully-actuated body capable of controlling attitude and translation separately. This effect is presented by the red curves on Figure 24, where roll and pitch are close to zero while the spatial trajectory is fulfilled according to Figure 25. Figure 23 presents the control allocation for this simulation case.

5.2.3 Case 3: Hexarotor of differently-sized propellers

The last case shows the ability of the algorithm to deal with different types of rotors in one multirotor by forcing maneuverability with $W_m = 1$ and $W_a = 0$. The architecture used here was proposed so as to increase multirotor efficiency and flight time (BLAIN, 2017). This approach considers two large rotors to provide most of the thrust for hovering while four smaller rotors are responsible for tilting the multirotor. The algorithm is capable of automatically assigning rotor speeds for this purpose considering the right rotor parameters are provided. The multirotor completes a 360° turn in this simulation.

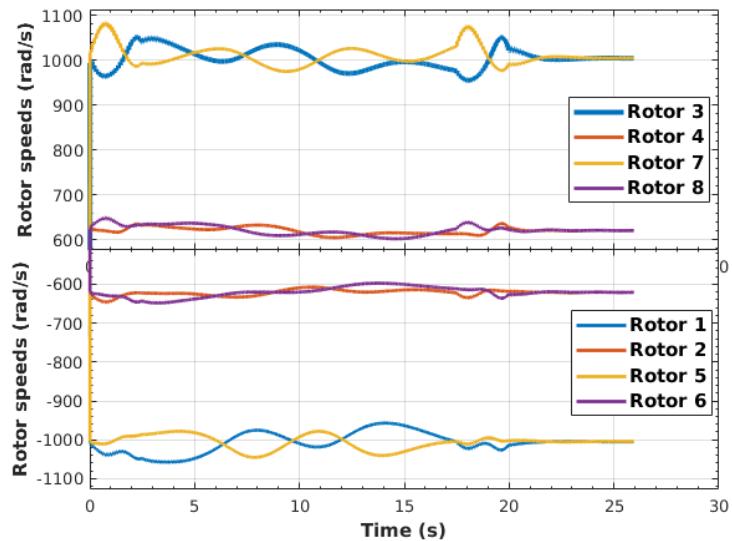


Figure 23: Rotor speeds allocation for $W_a = 1.2 \cdot 10^{11}$ in Case 2.

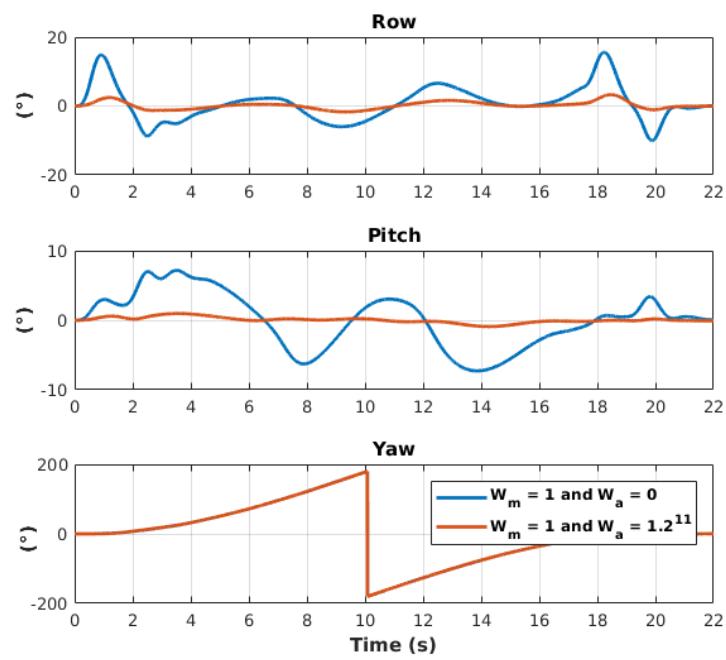


Figure 24: Attitude comparison for when $W_a = 0$ and $W_a = 1.2 \cdot 10^{11}$ in Case 2.

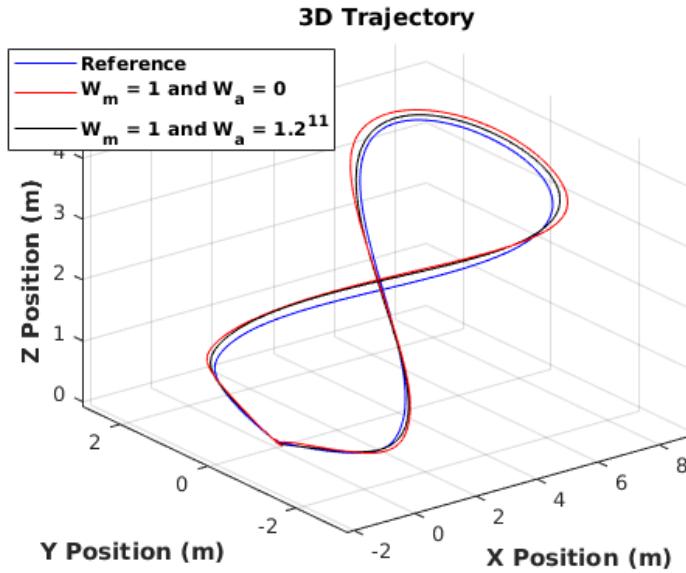


Figure 25: 3D trajectory for Case 2 simulations.

Figure 26 shows the difference of allocation for rotors 5 and 6, which are the larger ones. They remain practically at a constant speed providing lift while the other four rotors change speeds to accommodate the trajectory following.

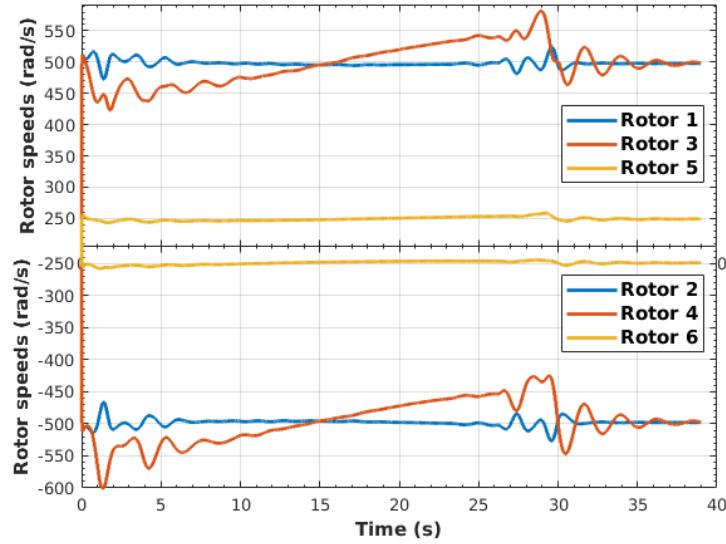


Figure 26: Control allocation for Case 3.

5.3 Controller optimization

38 optimizations were performed in order to prepare the presented controllers for comparison when it comes to their robustness to rotor failures and some parameter variations. Final values for each best optimized individual are presented on Appendix E. Figures 27 and 28 present the evolution of the best individual per generation for passive

controllers and active controllers, respectively. Table 20 summarizes the best individuals' fitness at the end of controller optimization as proposed on Section 4.4.2.

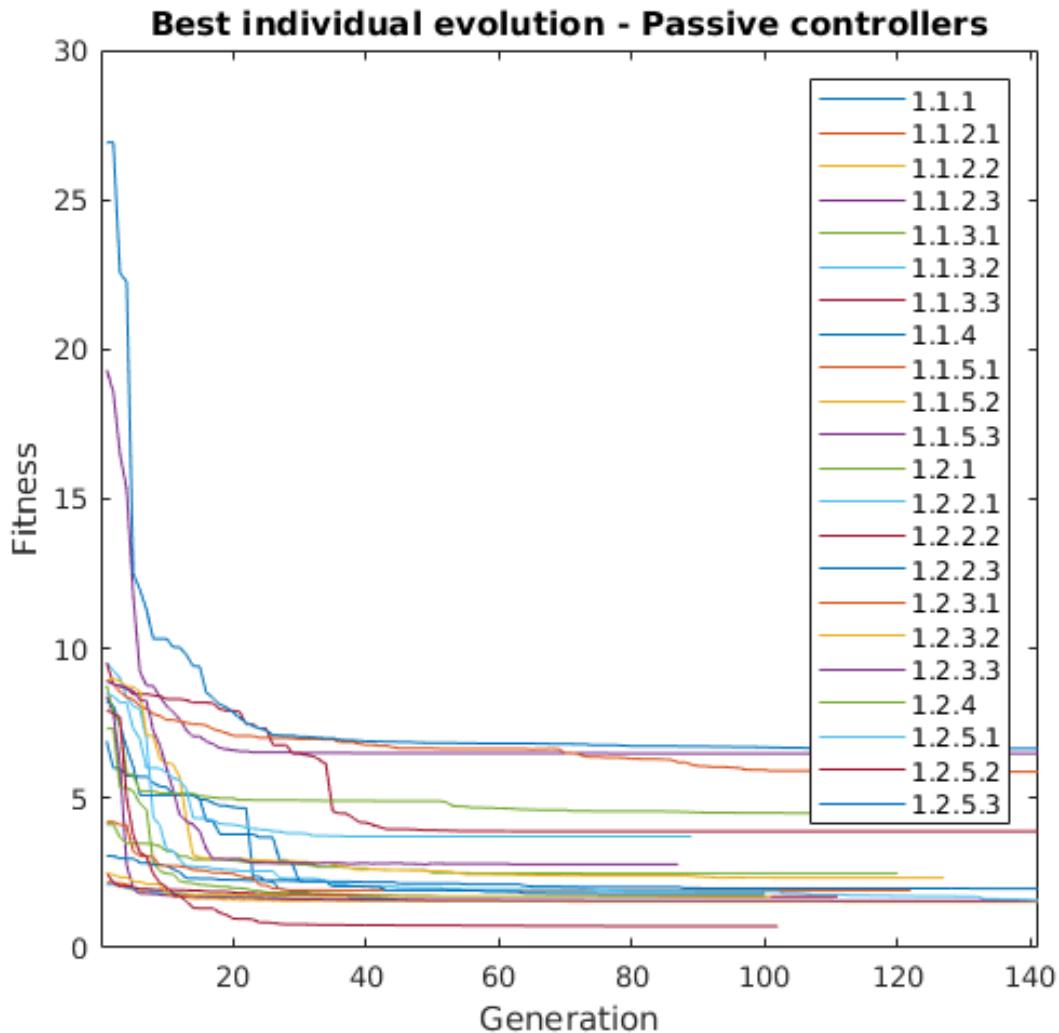


Figure 27: Best individual evolution per generation for passive controllers.

5.4 Controller evaluation

5.4.1 Samples distribution

Figures 29 to 30 present histograms for factors X_1 to X_7 , according to the distributions on Table 18, for both matrices M_1 and M_2 .

5.4.2 Passive Cases

This Section will present evaluation results for all passive cases. Model 1 will be presented first for later comparison with Model 2, which is more realistic.

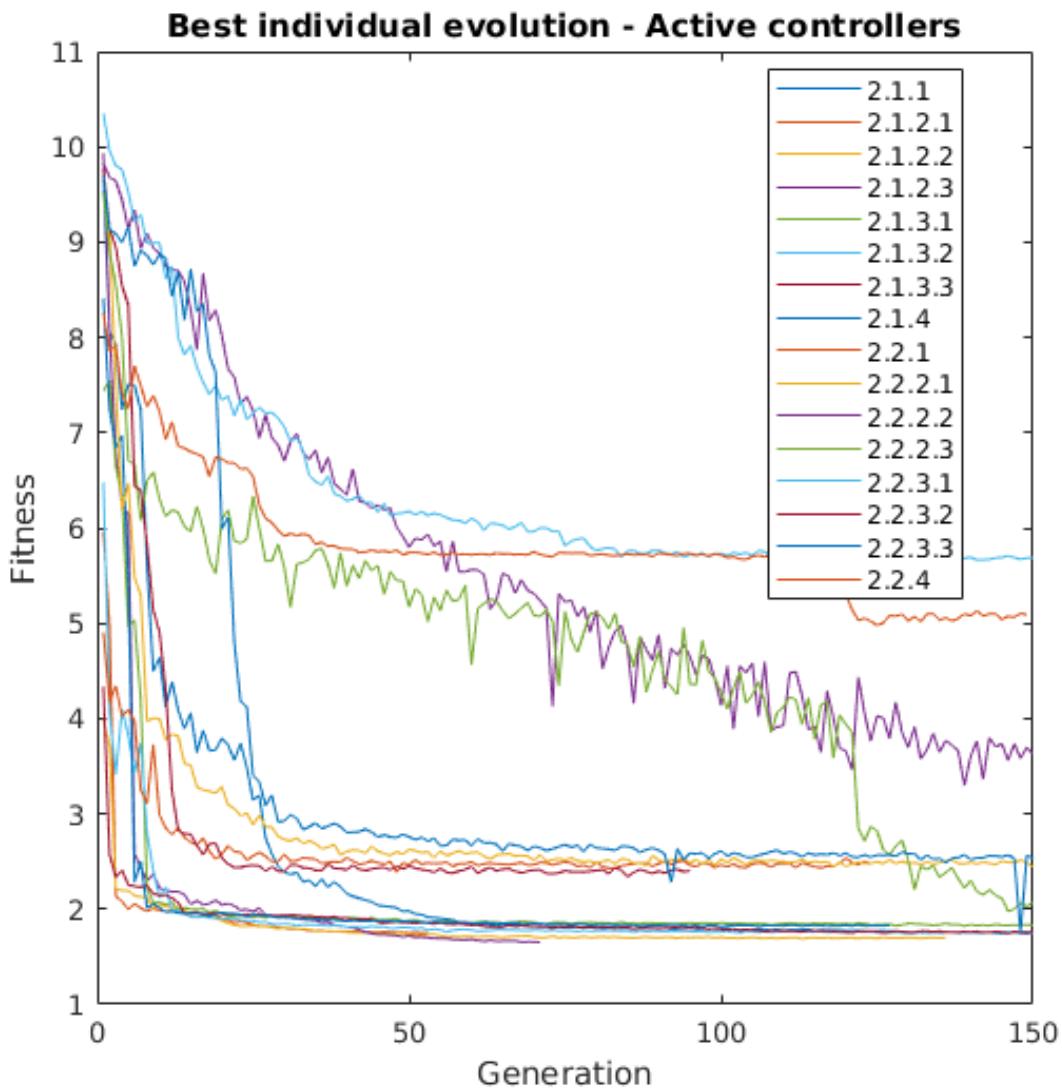


Figure 28: Best individual evolution per generation for active controllers.

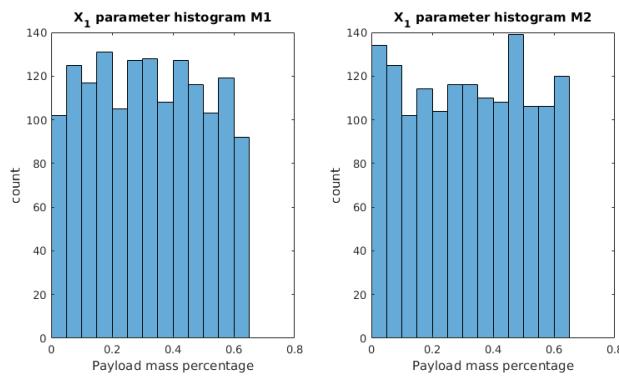
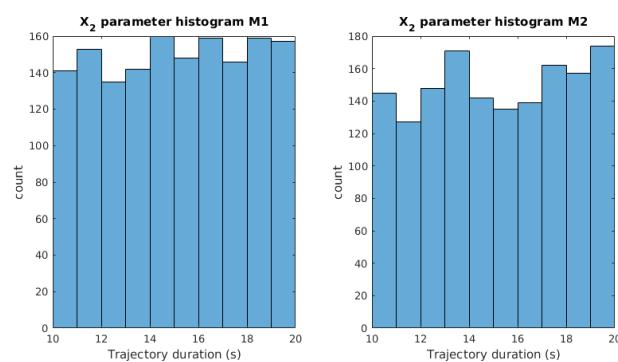


Figure 29: Histogram for factor X_1 used in Monte-carlo simulations.

Table 20: Summary of best fitness optimization

Case	Best fitness	Case	Best fitness
1.1.1	1.577	2.1.1	1.750
1.1.2.1	1.547	2.1.2.1	0.748
1.1.2.2	1.537	2.1.2.2	1.700
1.1.2.3	1.567	2.1.2.3	1.653
1.1.3.1	1.696	2.1.3.1	1.833
1.1.3.2	1.704	2.1.3.2	1.754
1.1.3.3	1.716	2.1.3.3	1.763
1.1.4	1.933	2.1.4	1.831
1.1.5.1	1.921	-	
1.1.5.2	1.723	-	
1.1.5.3	6.497	-	
1.2.1	2.484	2.2.1	2.478
1.2.2.1	1.579	2.2.2.1	2.444
1.2.2.2	3.892	2.2.2.2	3.624
1.2.2.3	1.810	2.2.2.3	2.069
1.2.3.1	5.816	2.2.3.1	5.687
1.2.3.2	2.337	2.2.3.2	2.402
1.2.3.3	2.791	2.2.3.3	2.529
1.2.4	4.499	2.2.4	5.088
1.2.5.1	3.719	-	
1.2.5.2	0.725	-	
1.2.5.3	6.649	-	

Figure 30: Histogram for factor X_2 used in Monte-carlo simulations.

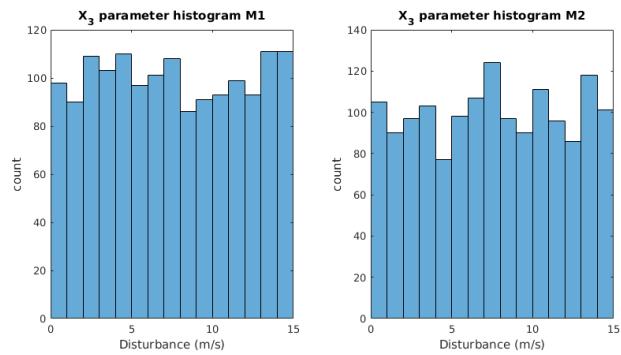


Figure 31: Histogram for factor X_3 used in Monte-carlo simulations.

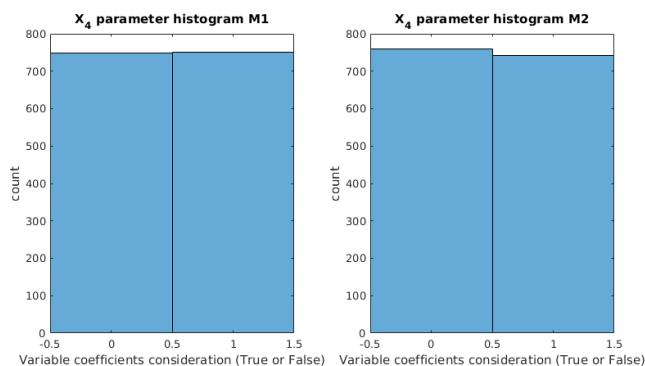


Figure 32: Histogram for factor X_4 used in Monte-carlo simulations.

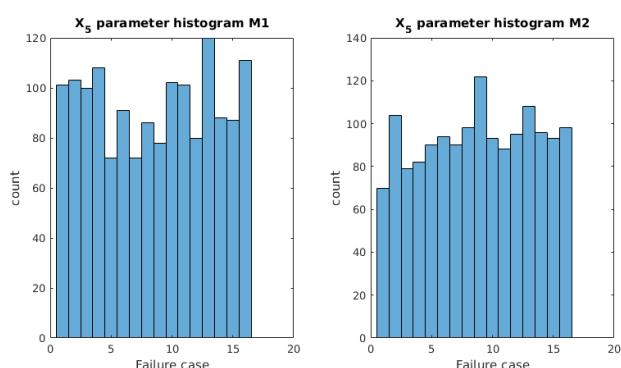


Figure 33: Histogram for factor X_5 used in Monte-carlo simulations.

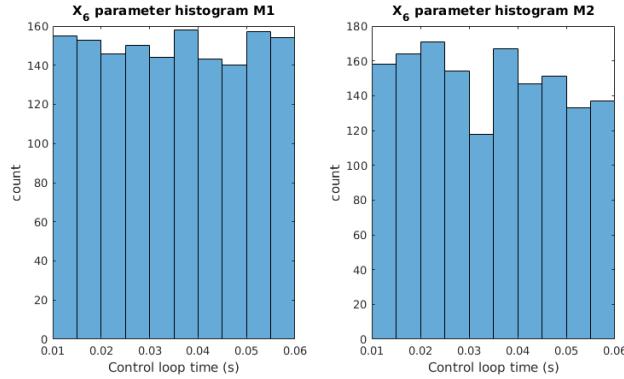


Figure 34: Histogram for factor X_6 used in Monte-carlo simulations.

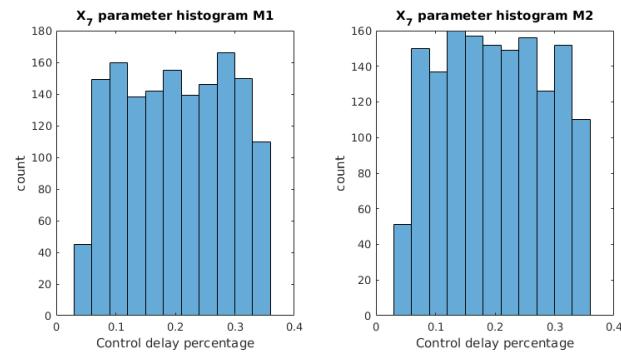


Figure 35: Histogram for factor X_7 used in Monte-carlo simulations.

5.4.2.1 Model 1

Table 21 presents factor-independent robustness metrics for Model 1 Passive cases. Table 22 summarizes performance metrics for the same controller cases. Sensitivity indices can be obtained on Appendix F.

5.4.2.2 Model 2

Table 23 presents factor-independent robustness metrics for passive controllers for Model 2. Table 24 presents Total effect and Sensitivity indices for simulation success, while Tables 25 and 26 present the same for position error and RMS power, respectively. Table 27 summarizes performance metrics for the same controller cases.

5.4.2.3 Model 2 best cases

From results presented on Section 5.4.2.2, the 5 chosen for a final comparison are 1.2.1 (PID), 1.2.2.3 (SMC-D), 1.2.3.2 (FT-LQR), 1.2.4 (PM-LQR) and 1.2.5.2 (MRAC-P). Figure 36 compares the five controller cases for their factor-independent robustness metrics in a spider graph. Figures 37, 38 and 39 do the same for the total effect indices for simulation success, position error and RMS power. Lastly, Figure 40 compares the five controllers when it comes to their performance after stability is guaranteed. Metrics in

Table 21: Factor-independent robustness metrics for Model 1 Passive Cases

Case	S_r	$E(p_{sim})$	$\hat{V}_{p_{sim}}$	$\hat{V}_{e_{p,RMS}}$	$\hat{V}_{P_{RMS}}$
1.1.1	0.5746	0.8261	0.06059	0.2611	$3.058 \cdot 10^5$
1.1.2.1	0.4043	0.6388	0.1246	0.4975	$4.85 \cdot 10^5$
1.1.2.2	0.4299	0.6662	0.117	0.4454	$4.405 \cdot 10^5$
1.1.2.3	0.3482	0.6482	0.1028	0.4812	$7.674 \cdot 10^5$
1.1.3.1	0.3581	0.6242	0.126	0.5123	$2.518 \cdot 10^5$
1.1.3.2	0.3672	0.6739	0.1043	0.3359	$3.027 \cdot 10^5$
1.1.3.3	0.459	0.7407	0.08799	0.3551	$4.943 \cdot 10^5$
1.1.4	0.2516	0.6394	0.0959	0.2839	$1.789 \cdot 10^5$
1.1.5.1	0.09593	0.2975	0.0893	5.694	$5.001 \cdot 10^5$
1.1.5.2	0.07281	0.2663	0.07867	2.603	$7.136 \cdot 10^5$
1.1.5.3	0	0.0553	0.0001266	9.895	$1.225 \cdot 10^6$

Table 22: Performance metrics for Model 1 Passive Cases

Case	$E(e_p)$	$E(V(e_p))$	$E(e_{att})$	$E(V(e_{att}))$	$E(P)$	$E(V(P))$	J_c
1.1.1	0.08098	0.004101	0.05687	0.06325	446.9	26211.0	0.001881
1.1.2.1	0.06651	0.002313	0.01293	0.01503	445.1	16822.0	0.002586
1.1.2.2	0.07136	0.003697	0.01458	0.01827	491.6	25177.0	0.004097
1.1.2.3	0.08217	0.002722	0.02913	0.01987	440.8	14866.0	0.009806
1.1.3.1	0.08418	0.003296	0.03786	0.04814	468.1	21466.0	0.01229
1.1.3.2	0.1003	0.006979	0.07315	0.07733	489.0	42699.0	0.01737
1.1.3.3	0.1119	0.005418	0.04911	0.05361	462.1	23800.0	0.01985
1.1.4	0.1828	0.02536	0.115	0.09693	468.8	40300.0	0.07147
1.1.5.1	0.1239	0.005507	0.09226	0.05798	447.7	16000.0	0.006716
1.1.5.2	0.09731	0.004929	0.05868	0.0415	429.0	17233.0	0.008308
1.1.5.3	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figures 36 and 40 are normalized in the sense that 1 represents the worst case and 0 the best one.

5.4.3 Active Cases

This Section will present evaluation results for all active cases.

5.4.3.1 Model 1

Table 28 presents factor-independent robustness metrics for Model 1 Active cases while Table 29 summarizes performance metrics for the same controller cases. Specific sensitivity indices can be obtained on Appendix F.

Table 23: Factor-independent robustness metrics for Model 2 Passive Cases

Case	S_r	$E(p_{sim})$	$\hat{V}_{p_{sim}}$	$\hat{V}_{e_{p,RMS}}$	$\hat{V}_{P_{RMS}}$
1.2.1	0.5355	0.8743	0.02639	0.1354	$1.417 \cdot 10^5$
1.2.2.1	0.1161	0.2977	0.08762	0.2612	$5.804 \cdot 10^5$
1.2.2.2	0.1768	0.6242	0.06776	0.1563	$4.542 \cdot 10^5$
1.2.2.3	0.1979	0.4902	0.1076	0.3652	$2.125 \cdot 10^5$
1.2.3.1	0.0000	0.1871	0.002506	0.1138	$3.544 \cdot 10^5$
1.2.3.2	0.1847	0.4638	0.118	0.2152	$2.398 \cdot 10^5$
1.2.3.3	0.0667	0.3482	0.07551	0.113	$1.381 \cdot 10^5$
1.2.4	0.0277	0.4738	0.0404	0.06618	$1.872 \cdot 10^5$
1.2.5.1	0.0057	0.3352	0.02562	3.917	$1.238 \cdot 10^5$
1.2.5.2	0.0381	0.2925	0.05155	6.073	$1.403 \cdot 10^6$
1.2.5.3	0.0000	0.0468	0.0001672	13.13	$3.254 \cdot 10^6$

Table 24: System stability robustness metrics for Model 2 Passive Cases

Case	$\hat{S}_{T1,p_{sim}}$	$\hat{S}_{T2,p_{sim}}$	$\hat{S}_{T3,p_{sim}}$	$\hat{S}_{T4,p_{sim}}$	$\hat{S}_{T5,p_{sim}}$	$\hat{S}_{T6,p_{sim}}$	$\hat{S}_{T7,p_{sim}}$	$\hat{S}_{1,p_{sim}}$	$\hat{S}_{2,p_{sim}}$	$\hat{S}_{3,p_{sim}}$	$\hat{S}_{4,p_{sim}}$	$\hat{S}_{5,p_{sim}}$	$\hat{S}_{6,p_{sim}}$	$\hat{S}_{7,p_{sim}}$
1.2.1	0.057	0.17	0.067	0.16	0.41	0.44	0.25	0.23	0.06	0.12	-0.033	0.45	0.2	$-3.5 \cdot 10^{-3}$
1.2.2.1	0.21	0.027	$-5.1 \cdot 10^{-3}$	0.15	-0.067	0.77	0.14	0.093	0.025	0.018	$-3.8 \cdot 10^{-4}$	$2.9 \cdot 10^{-3}$	0.41	0.02
1.2.2.2	0.46	0.38	0.31	0.39	0.27	0.56	0.31	0.12	$6.7 \cdot 10^{-3}$	0.017	0.12	0.18	0.19	0.067
1.2.2.3	0.49	0.33	0.12	0.43	0.058	0.54	0.35	0.16	0.015	$-1.6 \cdot 10^{-3}$	0.13	0.018	0.13	0.024
1.2.3.1	0.42	0.78	0.33	0.35	0.24	0.5	0.3	$7.9 \cdot 10^{-3}$	0.31	0.13	0.076	-0.033	0.22	0.023
1.2.3.2	0.082	0.19	0.1	0.085	0.018	0.79	0.27	0.029	$-3.1 \cdot 10^{-4}$	$8.4 \cdot 10^{-3}$	-0.017	0.037	0.56	0.032
1.2.3.3	0.25	0.21	0.19	0.14	0.14	0.75	0.3	0.019	-0.01	0.016	0.02	0.025	0.65	0.056
1.2.4	0.37	0.36	0.41	0.45	0.3	0.59	0.42	0.1	0.036	0.023	0.14	0.16	0.1	0.12
1.2.5.1	0.12	0.51	0.19	0.092	0.049	0.6	0.22	0.071	0.13	0.083	0.1	$-5.2 \cdot 10^{-3}$	0.22	0.041
1.2.5.2	0.26	0.35	0.4	0.22	0.13	0.7	0.19	0.019	0.062	0.21	$8.7 \cdot 10^{-3}$	$-9.6 \cdot 10^{-3}$	0.32	0.035
1.2.5.3	0.35	0.52	0.083	0.31	0.083	0.78	0.3	-0.063	0.32	$8.9 \cdot 10^{-3}$	-0.057	$8.8 \cdot 10^{-3}$	0.34	-0.04

Table 25: Position error robustness metrics for Model 2 Passive Cases

Case	$\hat{S}_{T1,pos}$	$\hat{S}_{T2,pos}$	$\hat{S}_{T3,pos}$	$\hat{S}_{T4,pos}$	$\hat{S}_{T5,pos}$	$\hat{S}_{T6,pos}$	$\hat{S}_{T7,pos}$	$\hat{S}_{1,pos}$	$\hat{S}_{2,pos}$	$\hat{S}_{3,pos}$	$\hat{S}_{4,pos}$	$\hat{S}_{5,pos}$	$\hat{S}_{6,pos}$	$\hat{S}_{7,pos}$
1.2.1	0.36	0.29	0.2	0.17	0.61	0.29	0.18	0.076	0.12	0.073	0.068	0.22	0.1	0.015
1.2.2.1	0.28	0.25	-0.011	0.12	-0.015	0.81	0.34	$-7.8 \cdot 10^{-3}$	$5.3 \cdot 10^{-3}$	0.027	0.038	-0.032	0.28	-0.084
1.2.2.2	0.57	0.44	0.26	0.41	0.47	0.42	0.34	-0.023	0.16	0.052	$5.8 \cdot 10^{-3}$	0.019	0.27	$-7.3 \cdot 10^{-3}$
1.2.2.3	0.49	0.54	0.14	0.37	0.083	0.57	0.43	0.12	0.091	$-5.0 \cdot 10^{-3}$	0.12	$8.8 \cdot 10^{-3}$	0.07	-0.016
1.2.3.1	0.62	0.74	0.54	0.43	0.37	0.64	0.55	0.011	0.18	0.096	0.11	-0.047	0.34	-0.06
1.2.3.2	0.064	0.22	0.059	$5.9 \cdot 10^{-3}$	0.11	0.7	0.1	0.013	0.17	0.037	0.038	$-5.9 \cdot 10^{-3}$	0.51	0.095
1.2.3.3	0.047	0.11	0.1	-0.021	0.028	0.96	0.088	-0.021	0.11	-0.082	-0.035	-0.052	0.56	$1.3 \cdot 10^{-3}$
1.2.4	0.51	0.6	0.47	0.35	0.31	0.47	0.41	-0.19	0.19	0.014	0.044	-0.03	0.13	$3.4 \cdot 10^{-3}$
1.2.5.1	0.41	0.73	0.21	0.26	0.046	0.51	0.24	0.025	0.34	-0.015	0.038	$1.6 \cdot 10^{-3}$	0.075	0.05
1.2.5.2	0.24	0.33	0.23	0.18	0.15	0.72	0.21	$-8.0 \cdot 10^{-3}$	0.15	0.011	0.016	0.013	0.72	$8.8 \cdot 10^{-3}$
1.2.5.3	0.23	0.66	0.095	0.21	0.095	0.81	0.28	0.056	0.042	$4.4 \cdot 10^{-3}$	0.021	$4.4 \cdot 10^{-3}$	0.38	0.016

Table 26: RMS Power robustness metrics for Model 2 Passive Cases

Case	$\hat{S}_{T1,pow}$	$\hat{S}_{T2,pow}$	$\hat{S}_{T3,pow}$	$\hat{S}_{T4,pow}$	$\hat{S}_{T5,pow}$	$\hat{S}_{T6,pow}$	$\hat{S}_{T7,pow}$	$\hat{S}_{1,pow}$	$\hat{S}_{2,pow}$	$\hat{S}_{3,pow}$	$\hat{S}_{4,pow}$	$\hat{S}_{5,pow}$	$\hat{S}_{6,pow}$	$\hat{S}_{7,pow}$
1.2.1	0.45	0.33	0.29	0.32	0.64	0.46	0.34	0.11	0.06	0.035	0.086	0.28	0.1	0.032
1.2.2.1	0.34	0.31	-0.03	0.27	-0.057	0.8	0.4	0.024	-0.043	$4.6 \cdot 10^{-3}$	-0.014	$-9.0 \cdot 10^{-3}$	0.23	$-1.8 \cdot 10^{-4}$
1.2.2.2	0.43	0.32	0.27	0.36	0.49	0.63	0.39	0.059	0.1	0.013	0.035	0.061	0.26	$-2.1 \cdot 10^{-3}$
1.2.2.3	0.39	0.51	0.14	0.31	0.048	0.7	0.39	0.08	0.078	0.013	0.076	0.13	$2.4 \cdot 10^{-3}$	-0.014
1.2.3.1	0.26	0.36	0.058	0.21	-0.072	0.61	0.12	$7.9 \cdot 10^{-3}$	0.23	$-2.9 \cdot 10^{-3}$	0.04	$-5.9 \cdot 10^{-3}$	0.23	$-9.8 \cdot 10^{-3}$
1.2.3.2	0.098	0.22	0.062	0.092	0.076	0.75	0.21	0.012	0.099	$-5.2 \cdot 10^{-5}$	0.017	-0.02	0.34	0.1
1.2.3.3	0.14	0.14	0.12	0.089	0.066	1.0	0.21	$3.7 \cdot 10^{-3}$	0.094	-0.048	-0.011	-0.058	0.31	0.058
1.2.4	0.17	0.23	0.12	0.23	0.097	0.63	0.25	-0.024	0.081	$5.1 \cdot 10^{-3}$	0.049	-0.022	0.28	0.12
1.2.5.1	0.3	0.53	0.29	0.22	0.15	0.49	0.21	0.051	0.4	0.018	0.062	0.098	0.13	0.13
1.2.5.2	0.33	0.45	0.29	0.26	0.22	0.68	0.39	0.029	0.2	0.026	0.017	0.036	0.46	-0.02
1.2.5.3	0.34	0.79	0.085	0.3	0.085	0.8	0.33	0.059	0.14	$3.0 \cdot 10^{-3}$	0.02	$3.1 \cdot 10^{-3}$	0.16	0.06

Table 27: Performance metrics for Model 2 Passive Cases

Case	$E(e_p)$	$E(V(e_p))$	$E(e_{att})$	$E(V(e_{att}))$	$E(P)$	$E(V(P))$	J_c
1.2.1	0.4229	0.06698	0.1589	0.1127	461.2	$1.334 \cdot 10^5$	0.0028
1.2.2.1	0.2857	0.04413	0.1937	0.09545	585.6	$2.176 \cdot 10^5$	0.003391
1.2.2.2	0.4117	0.08772	0.0904	0.0541	513.9	$7.142 \cdot 10^5$	0.004967
1.2.2.3	0.2822	0.03723	0.2657	0.1449	615.8	$3.278 \cdot 10^5$	0.01062
1.2.3.1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1.2.3.2	0.2769	0.02797	0.1132	0.08328	497.2	$1.189 \cdot 10^5$	0.01847
1.2.3.3	0.4218	0.03696	0.1221	0.06737	502.9	$3.767 \cdot 10^5$	0.02077
1.2.4	0.54	0.101	0.07352	0.09473	465.0	$3.799 \cdot 10^5$	0.07225
1.2.5.1	0.2537	0.01657	0.1093	0.05825	341.3	44488.0	0.007739
1.2.5.2	0.07145	0.002539	0.06254	0.0348	519.1	$5.098 \cdot 10^5$	0.0092
1.2.5.3	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 28: Factor-independent robustness metrics for Model 1 Active Cases

Case	S_r	$E(p_{sim})$	$\hat{V}_{p_{sim}}$	$\hat{V}_{e_{p,RMS}}$	$\hat{V}_{P_{RMS}}$
2.1.1	0.6214	0.8782	0.02605	0.1149	45177.0
2.1.2.1	0.6268	0.8816	0.02829	0.1094	71577.0
2.1.2.2	0.599	0.8683	0.02982	0.1349	$1.026 \cdot 10^5$
2.1.2.3	0.4986	0.8394	0.02724	0.1535	$1.125 \cdot 10^5$
2.1.3.1	0.6374	0.8875	0.02441	0.09263	42366.0
2.1.3.2	0.6394	0.8855	0.02429	0.1084	48011.0
2.1.3.3	0.5873	0.8613	0.03077	0.1314	80177.0
2.1.4	0.6321	0.8795	0.02898	0.1178	56111.0

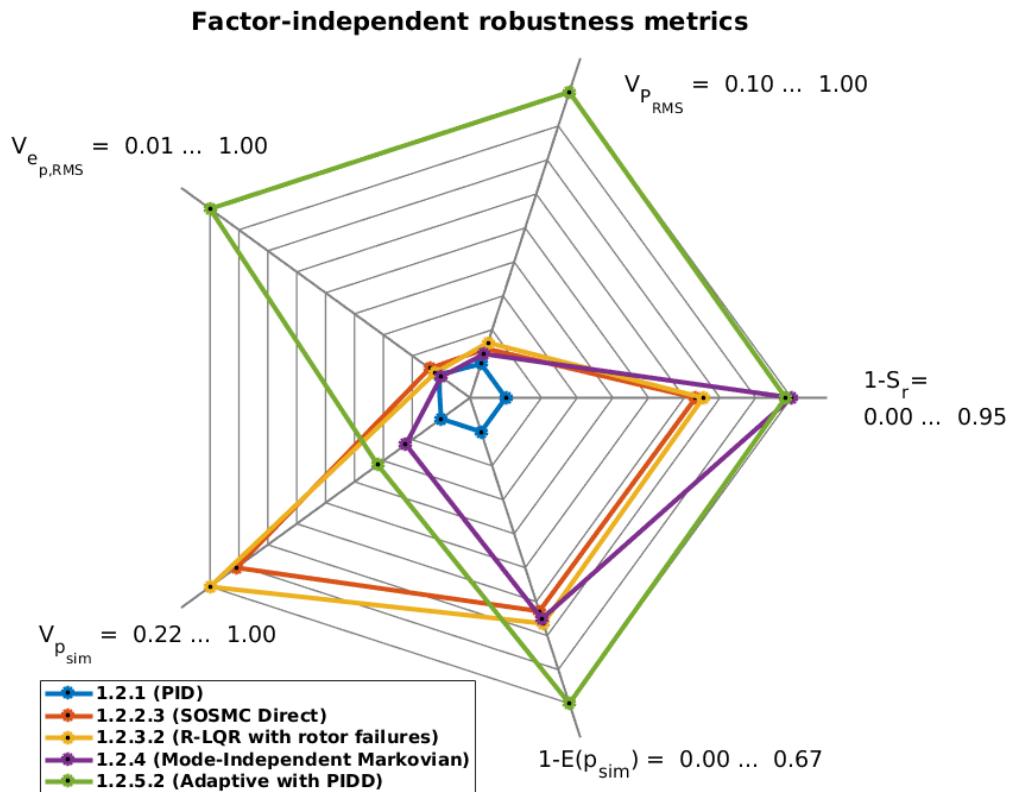


Figure 36: Factor-independent robustness metrics for best passive cases for Model 2.

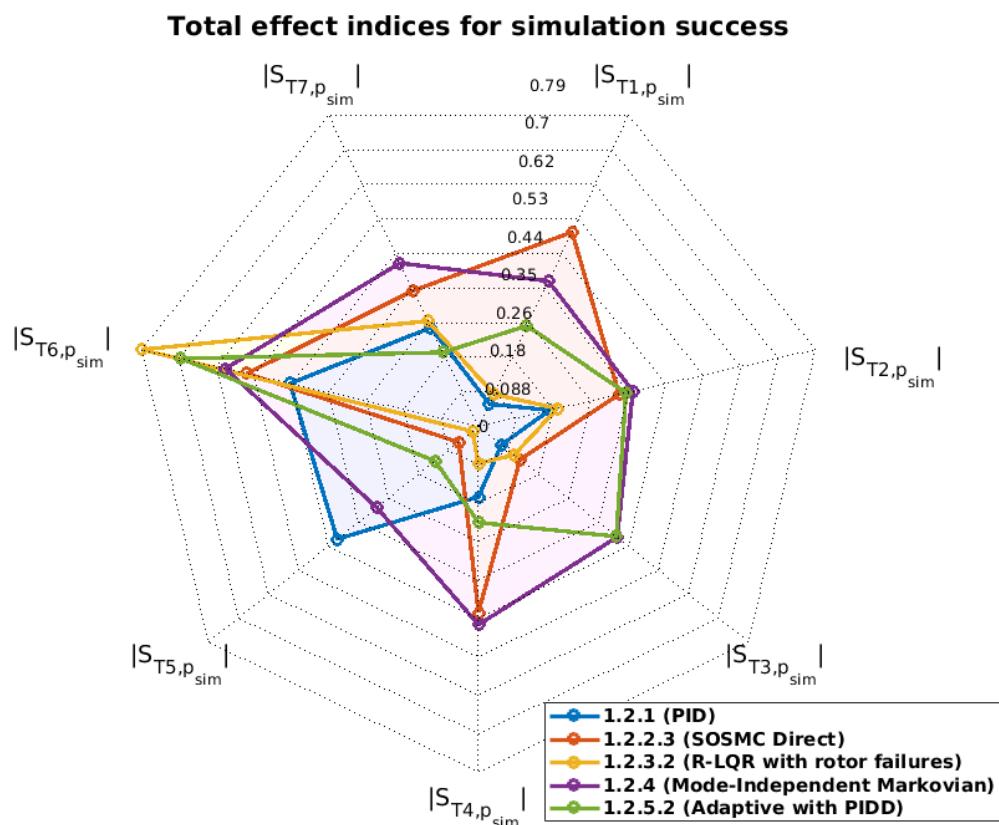


Figure 37: Total effect indices for simulation success in best passive cases for Model 2.

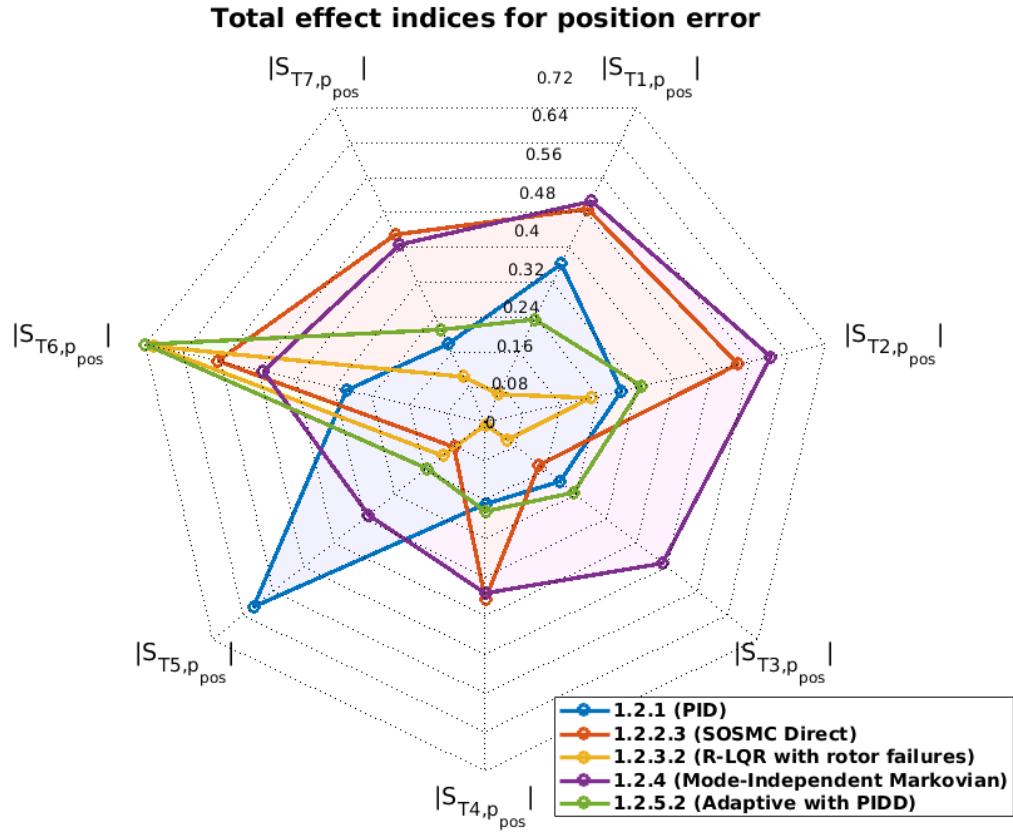


Figure 38: Total effect indices for position error in best passive cases for Model 2.

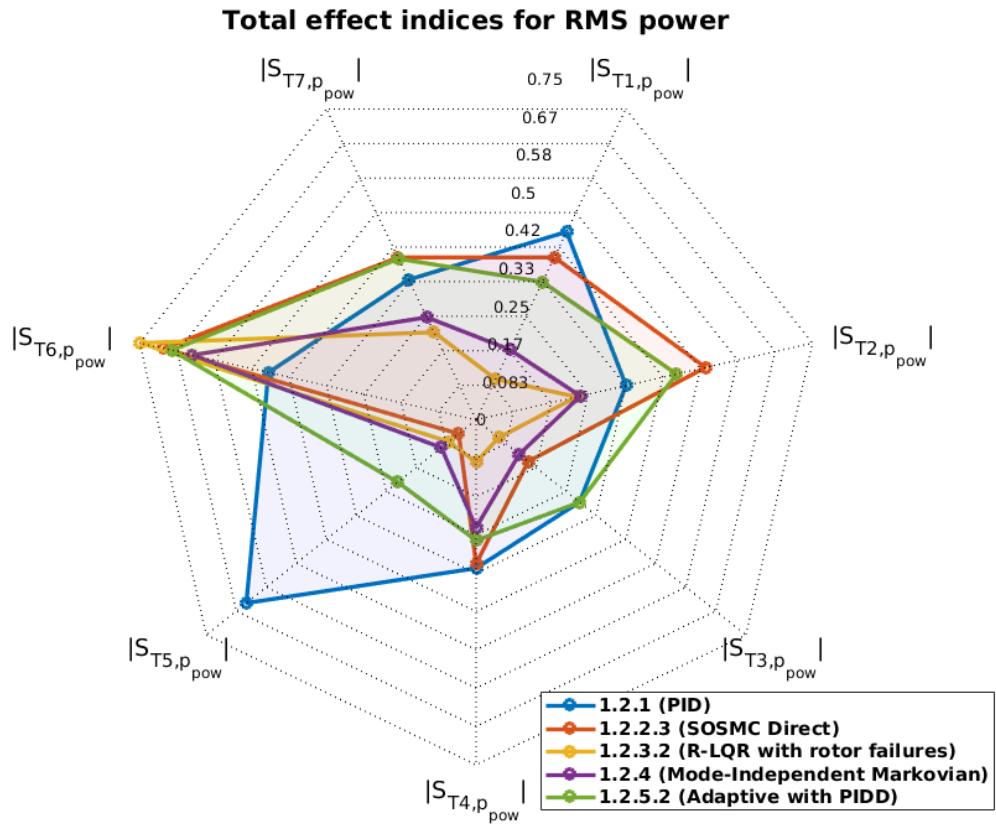


Figure 39: Total effect indices for RMS power in best passive cases for Model 2.

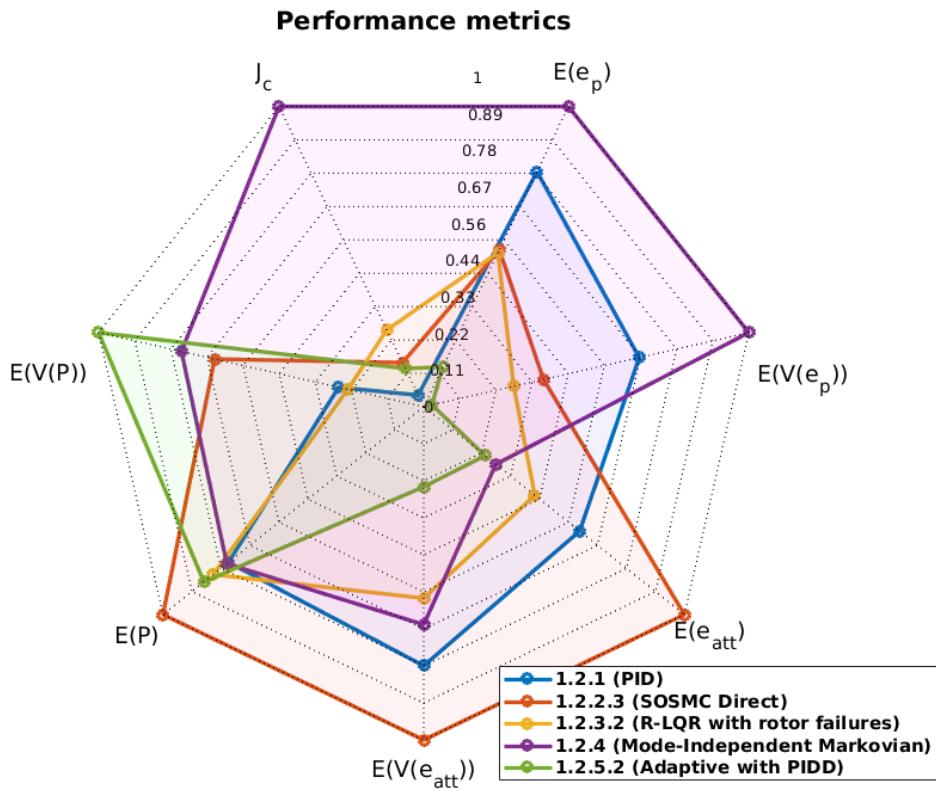


Figure 40: Performance metrics for best passive cases for Model 2.

Table 29: Performance metrics for Model 1 Active Cases

Case	$E(e_p)$	$E(V(e_p))$	$E(e_{att})$	$E(V(e_{att}))$	$E(P)$	$E(V(P))$	J_c
2.1.1	0.1142	0.01098	0.07746	0.07627	438.0	25033.0	0.01599
2.1.2.1	0.123	0.009504	0.03102	0.03748	442.2	35200.0	0.01694
2.1.2.2	0.09377	0.006546	0.02929	0.03311	435.8	24366.0	0.01847
2.1.2.3	0.114	0.009946	0.04704	0.0399	450.8	29644.0	0.01856
2.1.3.1	0.1524	0.0105	0.04624	0.06318	449.1	23200.0	0.02683
2.1.3.2	0.1279	0.005839	0.03148	0.05873	440.8	21533.0	0.0323
2.1.3.3	0.116	0.004733	0.0395	0.06439	442.2	24622.0	0.03468
2.1.4	0.1685	0.008432	0.05267	0.07791	458.7	25977.0	0.07609

Table 30: Factor-independent robustness metrics for Model 2 Active Cases

Case	S_r	$E(p_{sim})$	$\hat{V}_{p_{sim}}$	$\hat{V}_{e_{p,RMS}}$	$\hat{V}_{P_{RMS}}$
2.2.1	0.8060	0.9474	0.008304	0.0691	$1.021 \cdot 10^5$
2.2.2.1	0.3661	0.5582	0.1625	0.3327	$2.681 \cdot 10^5$
2.2.2.2	0.7227	0.9170	0.02015	0.09363	$2.202 \cdot 10^5$
2.2.2.3	0.2907	0.4830	0.1552	0.4476	$2.763 \cdot 10^5$
2.2.3.1	0.0000	0.1720	0.001796	0.05338	$1.328 \cdot 10^5$
2.2.3.2	0.5009	0.6866	0.1319	0.1886	$1.732 \cdot 10^5$
2.2.3.3	0.4945	0.6901	0.1373	0.276	$2.538 \cdot 10^5$
2.2.4	0.0106	0.2620	0.01597	0.04879	$3.967 \cdot 10^5$

Table 31: System stability robustness metrics for Model 2 Active Cases

Case	$\hat{S}_{T1,p_{sim}}$	$\hat{S}_{T2,p_{sim}}$	$\hat{S}_{T3,p_{sim}}$	$\hat{S}_{T4,p_{sim}}$	$\hat{S}_{T5,p_{sim}}$	$\hat{S}_{T6,p_{sim}}$	$\hat{S}_{T7,p_{sim}}$	$\hat{S}_{1,p_{sim}}$	$\hat{S}_{2,p_{sim}}$	$\hat{S}_{3,p_{sim}}$	$\hat{S}_{4,p_{sim}}$	$\hat{S}_{5,p_{sim}}$	$\hat{S}_{6,p_{sim}}$	$\hat{S}_{7,p_{sim}}$
2.2.1	-0.33	-0.24	-0.21	-0.44	1.2	-0.47	-0.46	0.11	0.025	-0.016	0.079	0.89	0.15	0.036
2.2.2.1	0.038	0.05	0.034	0.054	0.042	0.92	0.042	$7.6 \cdot 10^{-3}$	$-3.3 \cdot 10^{-3}$	0.016	$6.3 \cdot 10^{-3}$	0.024	0.9	-0.011
2.2.2.2	0.25	0.08	0.11	-0.028	0.44	0.12	0.11	0.11	0.077	0.044	0.19	0.84	0.31	-0.02
2.2.2.3	0.13	0.15	0.1	0.11	0.11	0.93	0.18	$8.9 \cdot 10^{-3}$	-0.011	$-2.0 \cdot 10^{-3}$	-0.01	0.011	0.69	-0.023
2.2.3.1	0.22	0.95	0.2	0.15	0.14	0.17	0.16	-0.024	0.57	0.15	0.013	0.017	0.051	$9.7 \cdot 10^{-3}$
2.2.3.2	0.029	0.034	0.041	0.036	0.14	0.94	0.16	0.021	0.015	$-8.2 \cdot 10^{-3}$	$1.5 \cdot 10^{-3}$	0.024	0.79	-0.021
2.2.3.3	0.062	0.099	0.081	0.053	0.11	0.95	0.21	0.012	$5.6 \cdot 10^{-3}$	$-3.1 \cdot 10^{-3}$	0.01	0.045	0.72	-0.025
2.2.4	0.16	0.16	0.8	0.18	0.17	0.39	0.22	0.02	0.03	0.51	0.018	0.026	0.23	$1.3 \cdot 10^{-3}$

Table 32: Position error robustness metrics for Model 2 Active Cases

Case	$\hat{S}_{T1,pos}$	$\hat{S}_{T2,pos}$	$\hat{S}_{T3,pos}$	$\hat{S}_{T4,pos}$	$\hat{S}_{T5,pos}$	$\hat{S}_{T6,pos}$	$\hat{S}_{T7,pos}$	$\hat{S}_{1,pos}$	$\hat{S}_{2,pos}$	$\hat{S}_{3,pos}$	$\hat{S}_{4,pos}$	$\hat{S}_{5,pos}$	$\hat{S}_{6,pos}$	$\hat{S}_{7,pos}$
2.2.1	0.33	0.48	0.26	0.19	0.62	0.17	0.1	0.086	0.15	0.12	-0.012	0.33	$7.4 \cdot 10^{-3}$	$-2.6 \cdot 10^{-3}$
2.2.2.1	-0.02	0.063	0.021	$1.8 \cdot 10^{-3}$	0.042	0.94	-0.043	0.018	0.067	$5.9 \cdot 10^{-3}$	$-5.3 \cdot 10^{-3}$	$1.7 \cdot 10^{-3}$	0.71	$-2.2 \cdot 10^{-3}$
2.2.2.2	0.46	0.49	0.31	0.34	0.74	0.47	0.22	0.048	0.07	0.12	-0.015	0.18	0.08	0.022
2.2.2.3	0.09	0.14	0.055	0.061	0.055	0.79	0.072	0.019	0.12	0.018	0.02	0.041	0.69	0.044
2.2.3.1	0.42	0.75	0.27	0.38	0.33	0.31	0.37	0.048	0.53	0.33	0.061	0.081	0.24	0.059
2.2.3.2	0.042	0.13	0.064	0.02	0.069	0.82	0.054	$7.8 \cdot 10^{-3}$	0.096	0.031	$-5.8 \cdot 10^{-4}$	0.12	0.68	0.051
2.2.3.3	0.016	0.1	0.02	$4.8 \cdot 10^{-4}$	0.047	0.86	$6.7 \cdot 10^{-3}$	$-9.1 \cdot 10^{-3}$	0.1	0.012	$2.0 \cdot 10^{-3}$	0.043	0.71	0.057
2.2.4	0.54	0.61	0.61	0.5	0.36	0.3	0.53	0.07	0.3	0.12	0.04	0.22	0.52	0.043

5.4.3.2 Model 2

Table 30 presents factor-independent robustness metrics for active controllers for Model 2. Table 31 presents Total effect and Sensitivity indices for simulation success, while Tables 32 and 33 present the same for position error and RMS power, respectively. Table 34 summarizes performance metrics for the same controller cases.

5.4.3.3 Model 2 FDD robustness

Table 35 presents variances for system stability, position error and power consumption for both FDD hit rate variation and FDD delay variation. Figures 41 and 42 shows simulation success variation graph for FDD hit rate and FDD delay variation. Appendix G

Table 33: RMS Power robustness metrics for Model 2 Active Cases

Case	$\hat{S}_{T1,pow}$	$\hat{S}_{T2,pow}$	$\hat{S}_{T3,pow}$	$\hat{S}_{T4,pow}$	$\hat{S}_{T5,pow}$	$\hat{S}_{T6,pow}$	$\hat{S}_{T7,pow}$	$\hat{S}_{1,pow}$	$\hat{S}_{2,pow}$	$\hat{S}_{3,pow}$	$\hat{S}_{4,pow}$	$\hat{S}_{5,pow}$	$\hat{S}_{6,pow}$	$\hat{S}_{7,pow}$
2.2.1	0.46	0.3	0.27	0.34	0.72	0.24	0.23	0.12	0.071	0.027	0.067	0.46	0.041	0.014
2.2.2.1	-0.056	$9.8 \cdot 10^{-3}$	-0.034	0.014	0.054	1.0	-0.065	0.065	0.052	$-5.7 \cdot 10^{-3}$	$-2.5 \cdot 10^{-3}$	0.021	0.63	0.024
2.2.2.2	0.46	0.39	0.2	0.25	0.85	0.4	0.14	$3.1 \cdot 10^{-3}$	0.031	0.038	-0.018	0.35	0.11	0.041
2.2.2.3	0.04	0.078	-0.051	0.06	$8.2 \cdot 10^{-3}$	0.86	0.026	0.013	0.093	0.031	$-9.6 \cdot 10^{-3}$	-0.02	0.75	0.046
2.2.3.1	0.067	0.15	$1.1 \cdot 10^{-3}$	0.096	-0.023	0.88	0.018	0.061	0.047	0.019	0.04	$5.0 \cdot 10^{-3}$	0.65	-0.01
2.2.3.2	-0.027	$-1.5 \cdot 10^{-3}$	-0.062	-0.025	0.16	0.87	-0.046	-0.018	$8.7 \cdot 10^{-3}$	-0.018	-0.013	0.18	0.6	0.068
2.2.3.3	-0.029	$-6.5 \cdot 10^{-3}$	-0.049	-0.025	0.12	0.89	-0.026	-0.013	0.061	$-7.7 \cdot 10^{-3}$	-0.026	0.11	0.58	0.066
2.2.4	-0.034	-0.048	-0.02	0.039	-0.038	1.0	0.051	0.029	$7.8 \cdot 10^{-3}$	0.014	$5.3 \cdot 10^{-3}$	$-2.3 \cdot 10^{-3}$	0.74	0.066

Table 34: Performance metrics for Model 2 Active Cases

Case	$E(e_p)$	$E(V(e_p))$	$E(e_{att})$	$E(V(e_{att}))$	$E(P)$	$E(V(P))$	J_c
2.2.1	0.3531	0.04548	0.1432	0.05051	495.5	$1.524 \cdot 10^5$	0.01748
2.2.2.1	0.2926	0.04039	0.1932	0.0898	468.7	$1.463 \cdot 10^5$	0.01873
2.2.2.2	0.4201	0.07153	0.09139	0.0689	503.1	$3.436 \cdot 10^5$	0.02003
2.2.2.3	0.2505	0.02613	0.2411	0.1806	530.1	$1.089 \cdot 10^5$	0.02006
2.2.3.1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2.2.3.2	0.4349	0.07738	0.09909	0.08403	489.1	$1.589 \cdot 10^5$	0.03391
2.2.3.3	0.3867	0.03877	0.108	0.06861	480.1	$2.236 \cdot 10^5$	0.03681
2.2.4	1.421	0.7073	0.08074	0.06784	444.5	$1.569 \cdot 10^5$	0.07725

Table 35: Model 2 robustness to FDD variation

Cases	FDD hit rate variation			FDD delay variation		
	$V(p_{sim})$	$V(e_{p,RMS})$	$V(P_{RMS})$	$V(p_{sim})$	$V(e_{p,RMS})$	$V(P_{RMS})$
2.2.1	0.0000	0.0002	$0.0569 \cdot 10^5$	0	$537.7 \cdot 10^{-6}$	33.760
2.2.2.1	0.0044	0.0070	$0.1110 \cdot 10^5$	0	$6.272 \cdot 10^{-6}$	15.120
2.2.2.2	0.1396	0.2916	$2.9620 \cdot 10^5$	0	$12.72 \cdot 10^{-6}$	0.1349
2.2.2.3	0.1167	0.2906	$2.5390 \cdot 10^5$	0	$3.047 \cdot 10^{-6}$	8.246
2.2.3.1	0.0000	0.0007	$0.1620 \cdot 10^5$	0	0.0000	0.0000
2.2.3.2	0.1414	0.1616	$1.358 \cdot 10^5$	0	$2.470 \cdot 10^{-6}$	0.0758
2.2.3.3	0.0000	0.0001	$0.5184 \cdot 10^5$	0	$37.67 \cdot 10^{-6}$	16.2000
2.2.4	0.0147	0.0883	$2.37 \cdot 10^5$	0	$1395.0 \cdot 10^{-6}$	0.0345

shows more graphs for position error and power variation to hit rate and delay variation.

5.4.3.4 Model 2 best cases

From results presented on Section 5.4.3.2, the 4 active cases chosen for a final comparison are 2.2.1 (PID), 2.2.2.2 (ASMC-P), 2.2.3.2 (AFT-LQR) and 2.2.4 (AM-LQR). Figure 43 compares the four controller cases for their factor-independent robustness metrics in a spider graph. Figures 44, 45 and 46 do the same for the total effect indices for simulation success, position error and RMS power. Lastly, Figure 47 compares the five

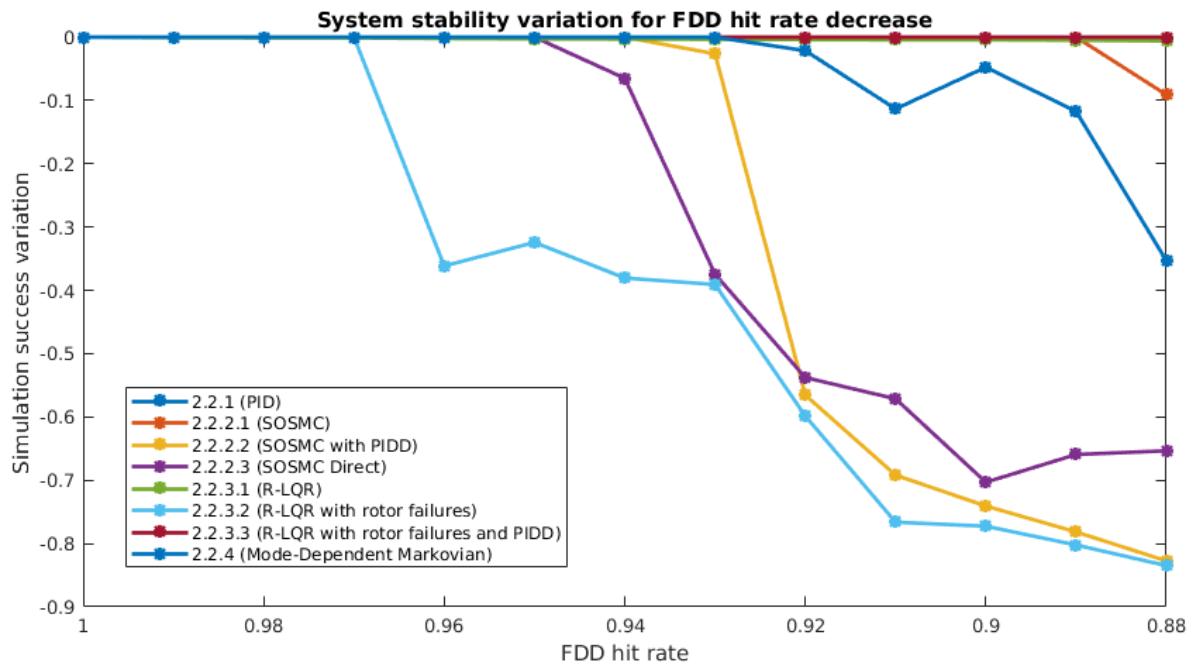


Figure 41: Simulation success variation to FDD hit rate decrease.

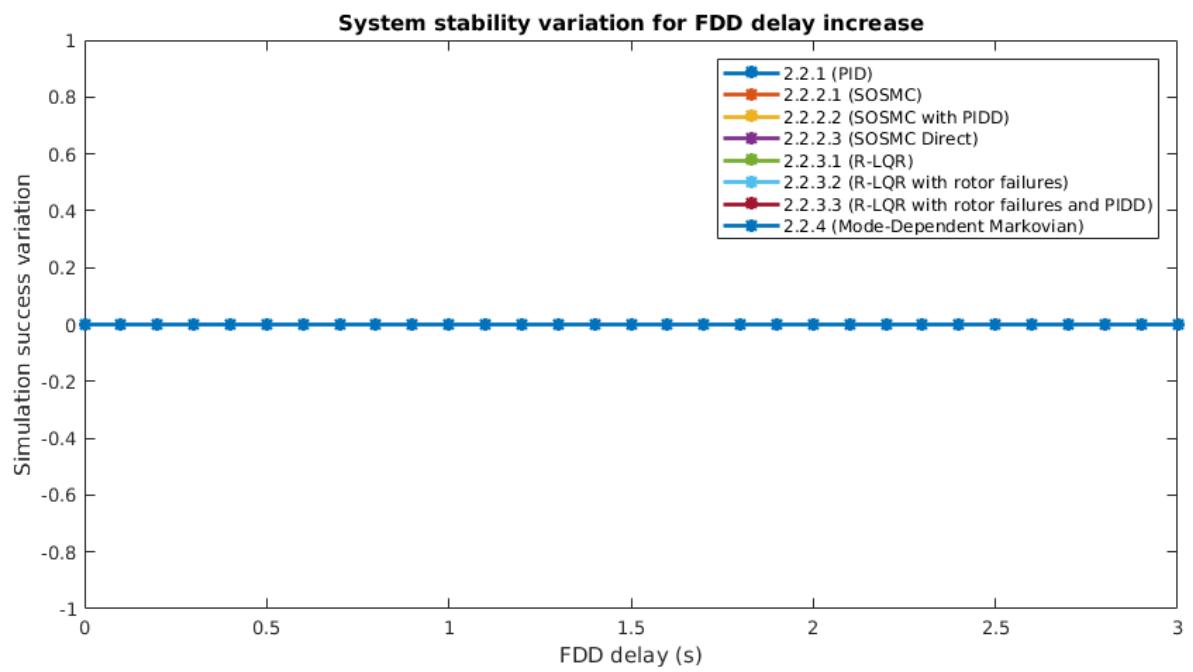


Figure 42: Simulation success variation to FDD delay increase.

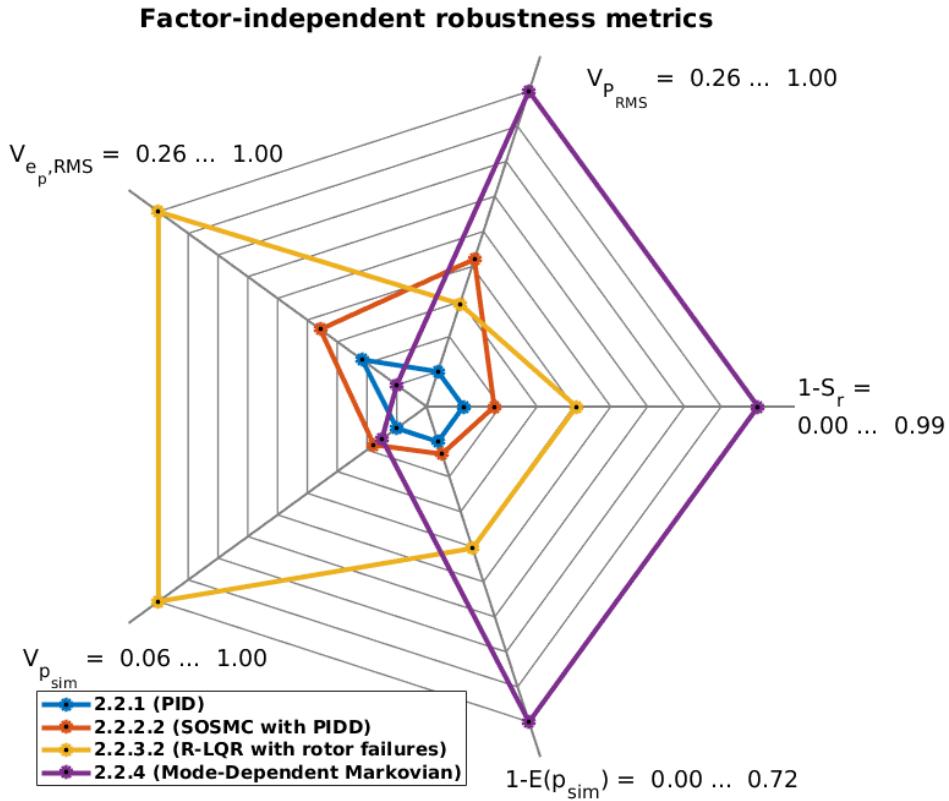


Figure 43: Factor-independent robustness metrics for best active cases for Model 2.

controllers when it comes to their performance after stability is guaranteed. Metrics in Figures 43 and 47 are normalized in the sense that 1 represents the worst case and 0 the best one.

5.4.4 Discussion

This Section will try to address all results presented before and their meanings. Each following Section will separately discuss results to better clarify them.

5.4.4.1 Difference between models

Tables 21, 28, 23 and 30 provide an overall comparison between simulating with Model 1 and Model 2. Cases with Model 1 are showed to provide better results compared to Model 2. Differences are very large particularly in the Adaptive and Markovian cases, where for Model 2 both controllers provide almost no robustness while in Model 1 they seem to perform well.

Two major factors may contribute to this difference: Model 1 does not consider motor dynamics and has a higher - not to say unrealistic - limit on rotor speeds. This causes the model to respond immediately to control demands. Moreover, considering hardware redundancy, the larger range of rotor speeds means that more energy is available

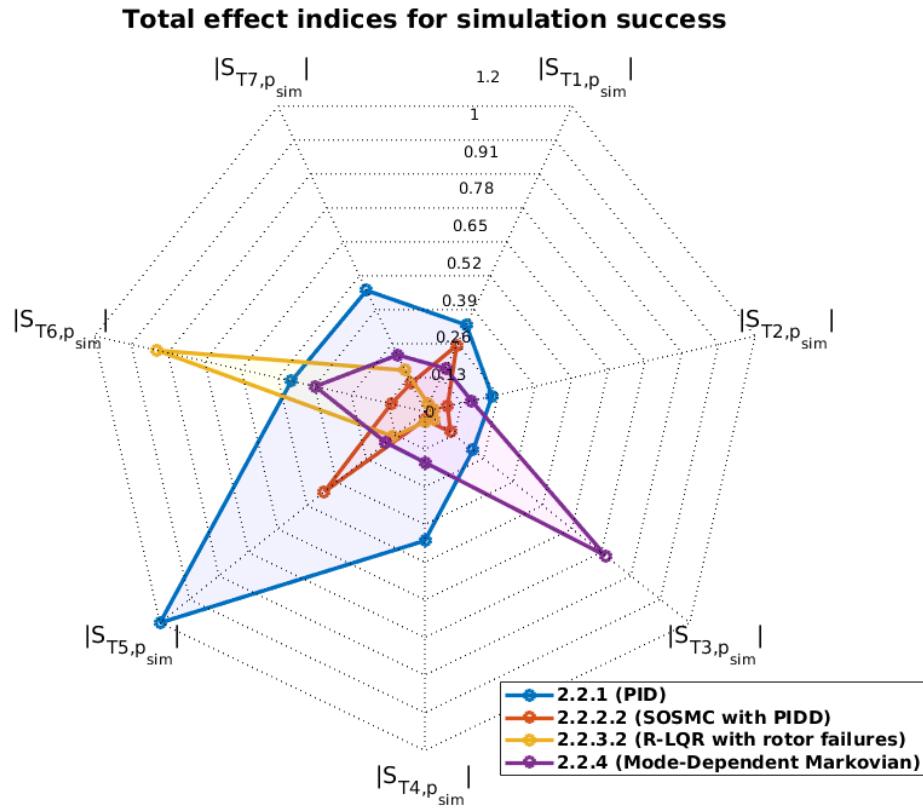


Figure 44: Total effect indices for simulation success in best active cases for Model 2.

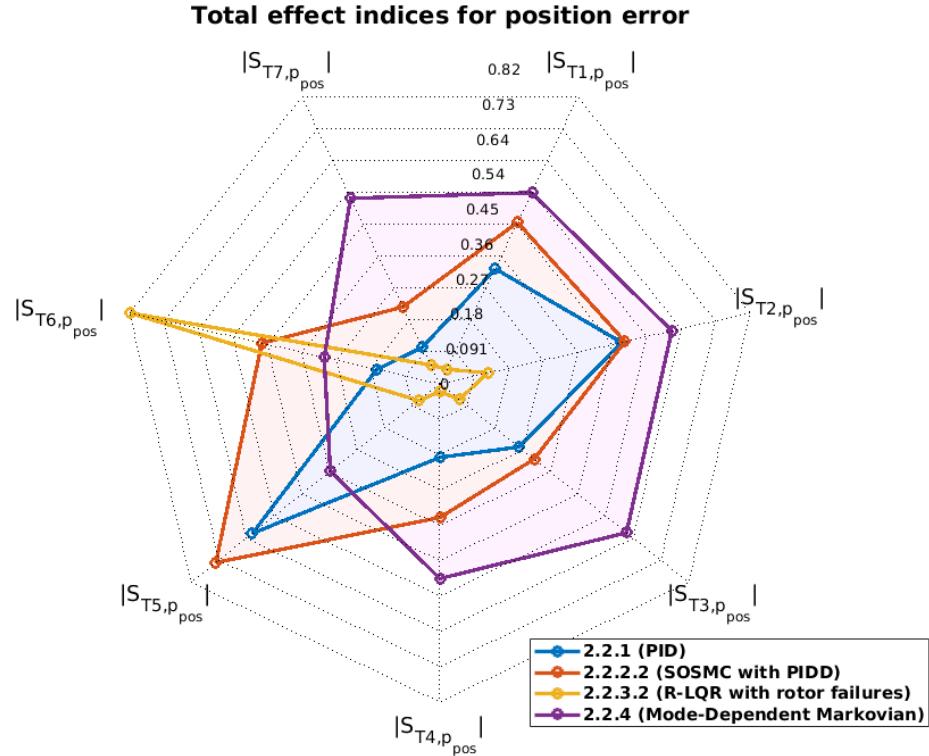


Figure 45: Total effect indices for position error in best active cases for Model 2.

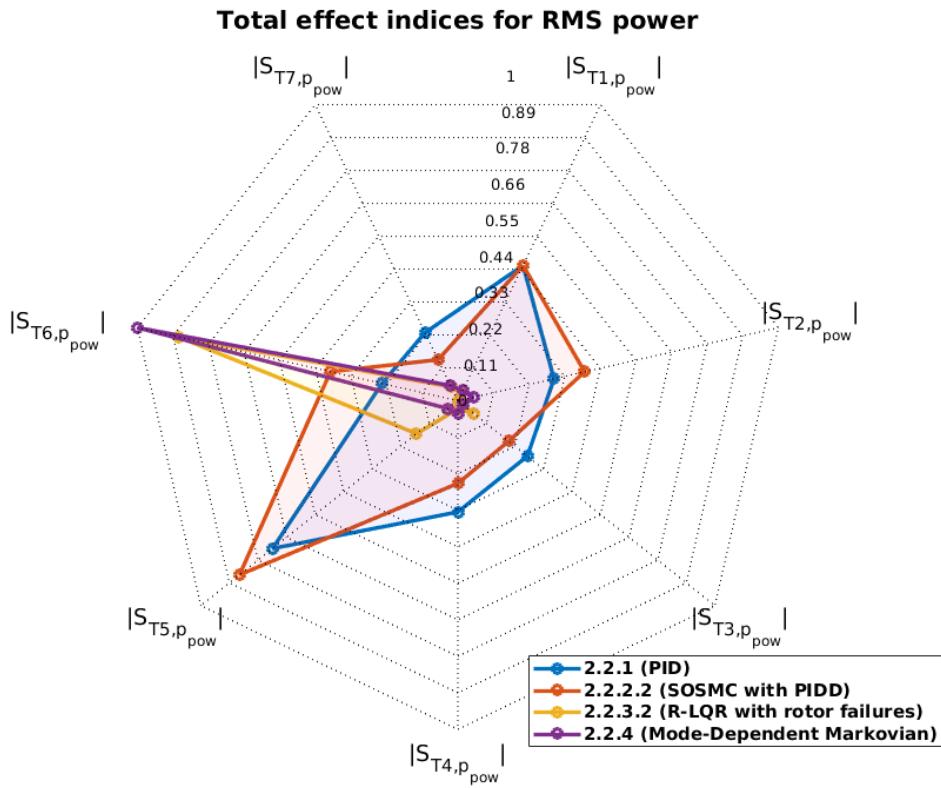


Figure 46: Total effect indices for RMS power in best active cases for Model 2.

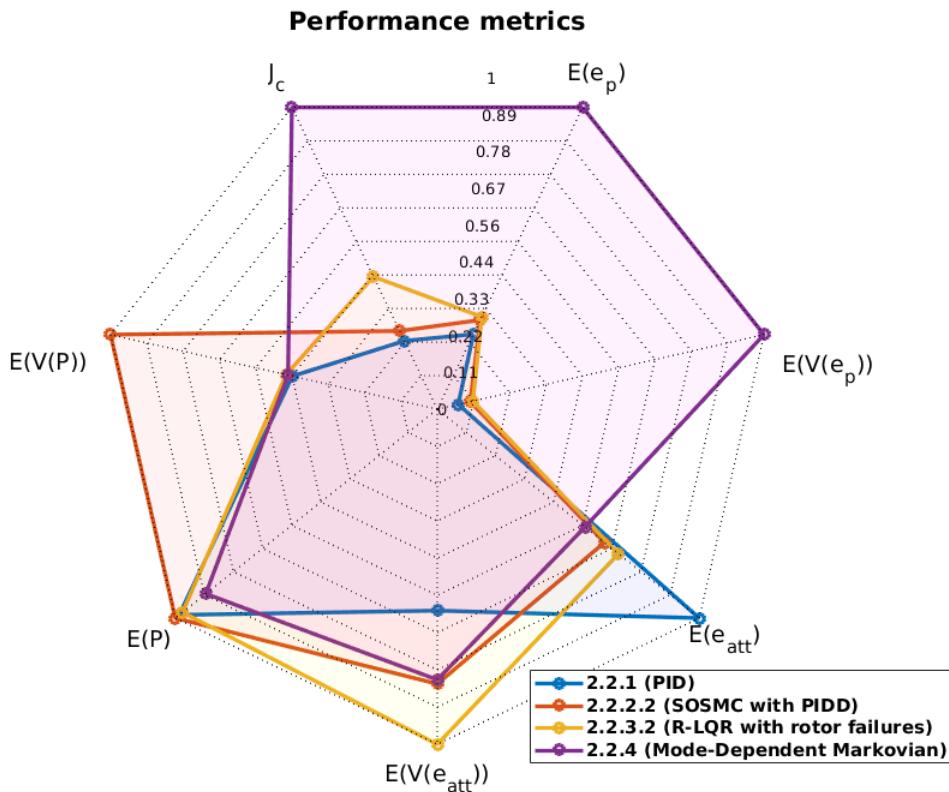


Figure 47: Performance metrics for best active cases for Model 2.

for the remaining rotors in cases of total failures, making rotor failure practically irrelevant to this model. It is important to note that the adoption of simplified models can lead to the erroneous conclusion about which controllers may be better than others when it comes to robustness and fault-tolerance, as by comparing Tables 21, 28, 23 and 30. One example is the Markovian case 2.1.4 which performs better than the PID case 2.1.1 for Model 1, as presented on Table 28, but performs too badly for the more realistic Model 2, as showed by Table 30. Another example is the better MRAC controller performance compared to the MRAC-P controller, as represented by cases 1.1.5.1 and 1.1.5.2 in Table 21. This is later showed to be not true when simulated with Model 2. Therefore, the comparison between Models 1 and 2 is a good showcase of how some factors cannot be left unconsidered when dealing with multirotors so as to guarantee a minimum relation to realistic scenarios.

Considering the presented above, all further discussions will take into consideration only simulations with Model 2, namely cases x.2.x.x.

5.4.4.2 SOSMC controllers

According to what was discussed before, (MERHEB; BATEMAN; NOURA, 2015) presented the controller represented on Section 3.4.2.1 without considering other factors such as motor dynamics, disturbances and payload. This work proposed the adaptations presented on Sections 3.4.2.3, 3.4.2.5, 3.4.2.2, 3.4.2.4 and 3.4.2.6, which could be compared with the original one under diverse situations as already discussed before.

One can analyse improvements and drawbacks of the modifications by looking at cases 1.2.2.1 (SSMC), 1.2.2.2 (SMC-P) and 1.2.2.3 (SMC-D) at Table 23. S_r indicates that the SMC-D is capable of completing more missions when under severe system variations compared to the SMC-P, which in case, is better than the SSMC also. $E(p_{sim})$, on the other hand, indicates that when the SMC-D fails, it does it relatively faster than the others, which is in opposition with the SMC-P. The latter is able to hold system stability longer when compared to the other two. Metrics $\hat{V}_{p_{sim}}$, $\hat{V}_{e_{p,RMS}}$ and $\hat{V}_{P_{RMS}}$ have to be analysed carefully. They have a sometimes dubious meaning: if $E(p_{sim})$ is low, higher values for $\hat{V}_{p_{sim}}$, $\hat{V}_{e_{p,RMS}}$ and $\hat{V}_{P_{RMS}}$ are good because it means the controller is able to at least for some simulation samples achieve higher stability, lower error and lower consumption; when $E(p_{sim})$ is high, the lower $\hat{V}_{p_{sim}}$, $\hat{V}_{e_{p,RMS}}$ and $\hat{V}_{P_{RMS}}$ the better, which means the controller does not reduce stability too much and does not increase error and consumption too much for the cases it gets unstable. Therefore, even though the SMC-P cannot finish more mission as the SMC-D, it is the most robust to system variations for having the largest $E(p_{sim})$ and smallest $\hat{V}_{p_{sim}}$ and $\hat{V}_{e_{p,RMS}}$, failing more when it comes to power consumption.

Tables 24 and 25 show that even though the modifications made the controller more robust, they became more sensitive to all parameter variations equally. The control loop period was the factor that most influenced the SSMC. With the modifications, factors

such as payload, speed, disturbances, lift and drag coefficients, and driver delay became more influent. It is interesting to note that rotor failures per se are the least influential factors to system stability, position error and power for SSMC and SMC-D. For Power consumption, as presented on Table 26, it is possible to observe that the control loop period remains as the most influential factor.

Both proposed modifications have slightly higher computational cost than the SSMC, as presented on Table 27, with the largest one being related to the SMC-D. Both also perform with higher chattering as can be implied from $E(V(P))$. The SMC-P, however, is able to maintain the desired attitude better than the other controllers by allowing higher position errors, while the SMC-D accomplish slightly lower position errors than the SSMC by allowing larger attitude errors.

When it comes to the active versions of the same controllers, S_r on Table 30 for cases 2.2.2.1, 2.2.2.2 and 2.2.2.3 indicates large improvements for all three controllers. The ASMC-D, however, is the one that presents the worst results in almost all metrics of Table 30. The ASMC-P, on the other hand, presents great advantage over the other two, being able to complete over 70% of simulated cases, while the ASMC accomplishes only 36% of stability success. The ASMC-P is also way more robust as previously discussed as can be noted by $\hat{V}_{p_{sim}}$, $\hat{V}_{e_{p,RMS}}$ and $\hat{V}_{P_{RMS}}$.

The use of FDD algorithms greatly reduces the sensitivity of the controllers, mainly for the ASMC and ASMC-D, as presented on Tables 31, 32 and 33. For the ASMC and ASMC-D, the main influencing factor by far for instability, position error and power consumption becomes the control loop period. Sensitivity for the ASMC-P is slightly more dispersed but with great prominence for the failure factor X_5 , which means that after holding stability for more than 70% of cases, some failure cases are the main factor for instability in the remaining 30%.

Table 34 indicates that the better robustness comes with a position error and chattering performance cost as indicated by $E(e_p)$ and $E(V(P))$, even though the ASMC-P is able to maintain a more stable attitude. The computational cost is similar for all three controllers.

Lastly, Table 35 indicates that to obtain the good fault-tolerant results for the ASMC-P, the FDD algorithm has to be very reliable so as to guarantee a high hit rate. This can be concluded from $V(p_{sim})$. The same applies to the ASMC-D. The FDD delay, in opposition, does not influence controller instability until the delay of 3 seconds, which may be a good time for fast systems like multirotors.

5.4.4.3 R-LQR controllers

(LEAO, 2015) was the reference for R-LQR controller applied to multirotor control. As discussed before, the absence of any actuator information in the system matrices was an indication for this dissertation's Author that this controller wouldn't be able to deal with rotor failures and large parameter variations. This hypothesis was confirmed by S_r of case 1.2.3.1 (R-LQR) on Table 23, which indicates the controller was not able to maintain stability over the range of all simulation cases. This result so degraded, however, seems to be a case of local minimum for the GA, which was not able to find a good individual even considering a large population of 3000 randomly distributed individuals. Another possibility is that the controller could be generating so large position errors and consumption that the fitness function identified finishing the simulation earlier as a means of reducing the fitness.

Modifications were proposed for this controller as presented on Sections 3.4.4.2, 3.4.4.3, 3.4.4.4, 3.4.4.5 and 3.4.4.6. Cases 1.2.3.2 and 1.2.3.3 in Table 23 indicate that the proposed modifications are able to cope with parameter variations and rotor failures to some extent, showing improvements over the "original" R-LQR. We shall refer to case 1.2.3.2 as FT-LQR (Fault-Tolerant-Robust-LQR) and to case 1.2.3.3 as FT-LQR-P.

Considering the R-LQR did not obtain stability in any case, its metrics are considered to be not valid and are not going to be analysed. S_r in Table 23 indicates that the FT-LQR achieves higher robustness to system variation when compared to the FT-LQR-P. $E(p_{sim})$ indicates that it can provide stability for longer times even in cases when it cannot finish the mission. The larger $\hat{V}_{p_{sim}}$ indicates lower robustness for system stability, which in this case of not so large S_r and $E(p_{sim})$ can be good meaning the controller can achieve some cases of almost mission complete.

Table 24 indicates that FT-LQR is not much sensitive to all factors, but stability is most influenced only by the control loop time and delay, followed by the trajectory speed (factor X_2). Factor influence in stability is more even for the FT-LQR-P, where the payload appears to be very influential also. Tables 25 and 26 maintains the same influences for position error and power but shows that the FT-LQR-P is less sensitive for diverse factors for position error and power when compared to stability, prevailing the same influences as for the FT-LQR.

Table 27 shows that the proposed FT-LQR has better performance compared to the FT-LQR-P, loosing only when it comes to attitude variance. This shows the potential of this controller type for dealing with both robustness and tracking performance. R-LQR performance metrics are presented as "NaN" because these metrics are calculated only for the cases of total complete trajectories or missions, which was not achieved by this controller.

When considering these controllers for active architectures, both proposed modifications present large robustness improvements with very similar results between them, as can be observed on Table 30. That is also the same even for factor sensitivities presented on Tables 31, 32 and 33, where the control loop time remains as the more influential and the rotor failures become the second most influential factor in stability, error and power. The R-LQR again presents no fault-tolerance and very low robustness to large system variations with no stable response at all.

Performances for active architectures for these controller cases, presented on Table 34, degrade when compared to the passive cases. They remain very similar between both proposed modifications, however. The difference between AFT-LQR and AFT-LQR-P relies on robustness to FDD variations, as presented on Table 35: The AFT-LQR gets unstable responses with higher FDD hit rates when compared to the AFT-LQR-P, which means it may not be the most suitable choice for when the FDD algorithm is not the most reliable one.

5.4.4.4 Adaptive controllers

(SCHWAGER; ANNASWAMY; LAVRETSKY, 2005) was the reference used for the Adaptive controllers. As cited before, (SCHWAGER; ANNASWAMY; LAVRETSKY, 2005) proposed an MRAC controller for fixed-wing aircrafts. Sections 3.4.3.1, 3.4.3.2 and 3.4.3.3 proposed the adaptation of the control structure for multirotor fault-tolerant control. The results for these controllers are presented as cases 1.2.5.1 (MRAC), 1.2.5.2 (MRAC-P) and 1.2.5.3 (MRAC-D) on Table 23. The MRAC-D does not present any satisfactory result and is then discarded for analysis. S_r indicates that the MRAC-P is able to complete more missions than the MRAC. $E(p_{sim})$ indicates, on the other hand, that the MRAC controller is slightly more robust for the simulation cases that it is not able to complete the trajectory.

Table 24 indicates that stability of the MRAC controller is very sensitive to the control loop time and the trajectory velocity, whereas for the MRAC-P, disturbances are more influential than the velocity. When it comes to position error, from Table 25, both controllers increase sensitivity to all factors almost evenly, with trajectory velocity becoming very influential in both controllers. The same applies for power consumption on Table 26.

The MRAC-P presents greater performance when compared to the MRAC controller, by comparing results on Table 27. It only loses in power consumption and computational cost, which is not much larger.

FDD algorithms were not considered for adaptive controllers since it would not make sense to use an adaptive approach if you can make use of explicit fault identification algorithms.

5.4.4.5 Comparison between best cases

After analysing proposed improvements for each controller architecture, it is necessary to compare architectures between each other. Figure 36 compares the best controllers from each architecture using the same factor-independent metrics presented before but normalized between them. Each axis of the radar plot represents a metric in such a way that the smaller the area taken by one controller, the more robust this controller is. In this sense, it is possible to visualize that the PID appears to be the most robust controller considering the parameter variations presented on Section 4.5.3. It is able to complete more missions than all other controllers and even in the cases where it gets unstable it is the one capable of maintaining stability longer, as suggested by the lowest $1 - E(p_{sim})$. The PID presents also the lowest variances, as suggested by $\hat{V}_{p_{sim}}$, $\hat{V}_{e_{p,RMS}}$ and $\hat{V}_{P_{RMS}}$. This may indicate that this controller provides a very similar position error and power consumption even under diverse system variations, until it gets unstable. All controllers but the Adaptive presented very similar position error and power consumption robustness. The Markovian and the Adaptive were the least stable, maintaining stability in many less cases when compared to the other three controllers. The Markovian, however, was as able as the SOSMC and the R-LQR to maintain stability longer before getting unstable. The lower stability robustness for the SOSMC and R-LQR, as suggested by $\hat{V}_{p_{sim}}$, indicates that they are able to achieve higher stability in some isolated mission cases when compared to the Markovian.

Figure 37 compares the Total-Effect indices for system stability for the best passive controllers, allowing the visualization of each controller sensitivity for each system factor or parameter. It is important to recast that the Total-effect index measures the influence of not only a parameter variation to system robustness, but also the influence of this parameter combined with all the others. In this sense, a low total-effect index for a parameter does not eliminate its influence to the system. This parameter may be rather "amplifying" the influence of another parameter that has a higher total-effect index. Considering this, it is notable that the control loop time is the most influential in destabilizing all controllers, as showed by $|S_{T5,p_{sim}}|$, with the PID being the least influenced while the R-LQR is the most influenced one. One can conclude from $|S_{T5,p_{sim}}|$ that the R-LQR and the SOSMC are the most fault-tolerant controllers when all other parameters are not taken into consideration. This means that, for settings where all system variations are well-controlled, these architectures may be the best ones for guaranteeing fault-tolerant stability. The R-LQR, along with the PID, is also very good when dealing with payload variations, with varied trajectory velocities and with disturbances, which may be good results for cargo transportation applications. The SMC-D, on the other hand, is very sensitive to payload and rotor coefficients variation, while the PID, surprisingly, seems to be much more sensitive to rotor failures when compared to all other controllers. This suggests that

the PID high success rate indicated by S_r may be higher than all others because it is capable of rejecting the influence of system parameter variations very well, facilitating stability when a failure occurs. The larger area for the Markovian indicates that there may exist a high coupling between all system parameters, which lead to a less successful result. The diverse failure models used in the Markovian, however, provided a lower fault-tolerance sensitivity to it when compared to the PID.

Controller position error sensitivities can be analysed on Figure 38. Graph areas for each controller remain relatively similar to the ones for system stability. Rotor failures and payload variations, however, become much more influential for the PID position error compared to their influence for PID stability, as suggested by $|S_{T1,p_{sim}}|$ and $|S_{T5,p_{sim}}|$. The SOSMC remains very sensitive to rotor coefficients variation, as suggested by $|S_{T4,p_{sim}}|$. This makes sense considering that this controller is much closer to rotor speed control when compared to the other ones since it does not make use of a CA. The R-LQR presents the smallest graph area, suggesting very low sensitivity to all parameters but the control loop period when it comes to position error.

When it comes to power consumption, the PID presents the largest graph area, suggesting that there's a high coupling between parameters that weakens the PID's power consumption robustness. The Markovian, in opposition, has its graph area reduced greatly, very similar to the R-LQR. All controllers but the R-LQR seem to have power consumption affected by rotor coefficient variations.

Figures 37, 38 and 39 suggest that if one maintains a very precise control loop period, the R-LQR have great chances of increasing robustness and fault-tolerance, maybe becoming more generally robust than the PID since influence from all other factors are smaller for almost all cases when compared to the PID.

The Adaptive controller showed to be the least robust when compared to all others, as presented above. If one fix the operation scenario, however, this architecture may be of interest considering its higher performance, as suggested by Figure 40, with a slight higher motor chattering as indicated by $E(V(P))$. The PID, on the other hand, presents the second worst position and attitude errors and error variances, as presented by $E(e_p)$, $E(V(e_p))$, $E(e_{att})$ and $E(V(e_{att}))$. This suggests higher aircraft oscillations when compared to the other controllers but the SOSMC. The SOSMC slight higher robustness is negatively compensated by its attitude and power consumption performances, which are presented to be the worst ones. The Markovian computational cost is notable when compared to all others, which may invalidate this control architecture for multirotors with low computational resources. The R-LQR controller shows very good performance even after guaranteeing relatively high robustness and fault-tolerance capabilities. This again suggests that this controller may be a good competitor to the PID considering the sensitivity remarks already presented, which would be able to cope with both robustness

and performance.

A lot changes when one adds FDD capabilities to the controllers. The best SOSMC controller becomes the ASMC-P, as already presented. Figure 43 indicates that its robustness becomes much closer to the PID's when compared to the passive case. The Markovian controller, on the other hand, unexpectedly degrades its robustness when compared to the passive case. The R-LQR, surpasses the SOSMC only in robustness for power consumption.

The even distribution among factors for the PID in Figure 44 along with the high S_r on Table 30 suggest that there's not much room for improvements in the PID controller at this point. The higher $S_{T5,psim}$ corroborates this by indicating that mission success may be improved mainly by reducing rotor failures, which does not make sense since fault-tolerance is desired. The R-LQR, however, presents almost nonexistent influence in system stability from payload, trajectory velocity, disturbances and rotor coefficients variation, with very low sensitivities to control delay and rotor failures. This leaves room for improvements once again in the control loop period and control delay, which are much more "controllable" factors from the view of controller design when compared to other factors.

The SOSMC controller presented unexpected results on Table 31: Its sensitivity indices are larger than its total-effect indices. This happened in few other cases, but only for very small index results, which could be regarded as estimator errors. In this particular case, values are relatively larger. This could indicate that factor interference may be helping the controller to increase robustness while the factors isolated are decreasing robustness. Other hypothesis is that factors may not be orthogonal for this case in opposition to the premisses of Section 4.5, as suggested by (SALTELLI et al., 2004). That is not a strong assumption, though, considering that it hasn't happened for any other case and this would be inherent to the factors probability distributions and not the controller itself.

In the case of position error sensitivity, Figure 45 presents even sensitivities for all controllers but the R-LQR, which shows high robustness to payload and rotor coefficients variations. For the PID, trajectory velocities become more influential, while the control delay appears to be a problem for the Markovian controller.

Payload variations emerge as high influential in power consumption robustness for both the SOSMC and the PID. For both the Markovian and R-LQR, only the control loop delay appears to have influence in power consumption, which would probably increase because of some chattering and not necessarily the mean value.

When it comes to performance of the active architectures, all controllers but the Markovian present very similar position error performances, with the PID performing slightly better, as in opposition of the passive case. The PID presents larger attitude error but with lower attitude oscillations, which may be a good approach in some transport

cases. The R-LQR presents the worst attitude oscillations in this case while the SOSMC presents higher chattering, as induced by $E(V(P))$. Mean power consumption, however, is very similar for all controllers.

As discussed before, the FDD delay presents no risk for all controllers in the cases investigated. The SOSMC and R-LQR, however, have to have special attention when being considered for active control since their stability, position error and power consumption is more easily affected by low FDD hit rates, as presented on Table 35 and Figure 41.

5.4.4.6 Markovian Robust Regulators

High computational cost and a large number of parameters for tuning are major drawbacks for the use of Markovian regulators in multirotor aircrafts. The possible number of faulty situations create a large pool of possible Markovian modes for modelling, and consequently requires operations with huge matrices. Putting these factors aside, one should not discard these controllers for aircraft fault-tolerant control.

Two factors may have caused the Markovian controllers to have very low robustness and performance results: In the first, matrix P in the Active Markovian was not optimized, in opposition to what was considered for the R-LQR controllers. Instead, it was adopted as the identity. This was a design choice to reduce the number of tuning parameters for the GA and try to facilitate finding a good controller among the large range of possible values. Optimizing matrix P , however, could facilitate the Riccati's convergence to an optimal value for better controller response. In the second, since each failure mode generates a different Markovian mode and hence a new set of matrices, it was not possible to consider the range of all failures. This may have led to worse results and could be avoided if all failure modes were considered, which would in consequence increase computational cost even more.

6 CONCLUSION AND FUTURE WORKS

This dissertation presented an in-depth comparative analysis of robust controllers applied to fault-tolerant control for hardware-redundant multirotors. First, a motivation for the study was presented, showing statistics related to transportation. Then a review on fault-tolerant control architectures was introduced, clarifying concepts related to the study subject. Attention was given to explain the importance of a well executed robustness analysis based on quantitative metrics from sensitivity analysis methods and systematic comparison. This was used to emphasize the innovative contribution of this work. A generalized multirotor model was presented, with fair modelling considerations to facilitate simulations but also not to leave important factors unmodelled. Much care was taken on contributing with an architecture-independent control allocation and attitude planning, which facilitated implementing failure simulations and fault-tolerant algorithms. Control architectures were then presented, indicating proposed modifications for improvements in fault-tolerance and robustness, mostly treating fault-tolerance by means of robustness.

All presented controllers were then optimized following a non-subjective genetic algorithm for posterior evaluation and comparison. Results presented showed the efficacy of the proposed control allocation and attitude planning methods. They also included the optimization results and evolution. Controller evaluations were compared by means of diverse metrics and radar plots, indicating the best controllers for each case. Sensitivity analysis was able to provide insights of where some controllers could have robustness improved by improving some design factors or limiting the range of operation. The use of two different models for simulation was able to show also the importance of modelling factors on the controller response analysis, which could lead to erroneous conclusions.

Future contributions to this work include the implementation of some controller cases and simulations in a real aircraft model, as presented on the Appendix, for validation. The multirotor model used here may also be improved to provide more realistic results after experiments. It is important to note that this work should be interpreted not only as limited to the control algorithms and optimizations presented here, but should be interpreted as the introduction of a systematic model-based approach for rigorous and realistic control algorithms development.

BIBLIOGRAPHY

Abu Rmilah, M. H. Y.; HASSAN, M. A.; Bin Mardi, N. A. A PC-based simulation platform for a quadcopter system with self-tuning fuzzy PID controllers. **Computer Applications in Engineering Education**, v. 24, n. 6, p. 934–950, 2016. ISSN 10613773. Disponível em: <<http://doi.wiley.com/10.1002/cae.21769>>.

AERONES. **Ultimate solution for most wind turbine problems**. 2019. Disponível em: <https://www.aerones.com/eng/wind_turbine_maintenance_drone/>. Acesso em: June 1st 2019.

AL, C. M. H. et. Commuting distance, cardiorespiratory fitness, and metabolic risk. **American Journal of Preventive Medicine**, v. 42, n. 6, p. 571–578, 2012.

ALAIMO, A. et al. Mathematical modeling and control of a hexacopter. In: **Unmanned Aircraft Systems (ICUAS), 2013 International Conference on**. [S.l.: s.n.], 2013. p. 1043–1050. ISBN 9781479908172.

_____. PID Controller Applied to Hexacopter Flight. **Journal of Intelligent & Robotic Systems**, v. 73, n. 1-4, p. 261–270, jan 2014. ISSN 0921-0296. Disponível em: <<http://link.springer.com/10.1007/s10846-013-9947-y>>.

ALWI, H.; EDWARDS, C. Fault tolerant control of an octorotor using LPV based sliding mode control allocation. In: **2013 American Control Conference**. IEEE, 2013. p. 6505–6510. ISBN 978-1-4799-0178-4. ISSN 21621195. Disponível em: <http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6580859http://ieeexplore.ieee.org/document/6693886/http://ieeexplore.ieee.org/document/65808>.

_____. Sliding mode fault-tolerant control of an octorotor using linear parameter varying-based schemes. **IET Control Theory & Applications**, v. 9, n. 4, p. 618–636, feb 2015. ISSN 1751-8644. Disponível em: <<http://digital-library.theiet.org/content/journals/10.1049/iet-cta.2014.0215>>.

ALWI, H.; HAMAYUN, M. T.; EDWARDS, C. An integral sliding mode fault tolerant control scheme for an octorotor using fixed control allocation. In: **13th International Workshop on Variable Structure Systems (VSS)**. [s.n.], 2014. v. 2, n. 2, p. 1–6. ISBN 978-1-4799-5566-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6881156>>.

ANTCLIFF, K. R.; GOODRICH, K.; MOORE, M. **NASA Silicon Valley Urban VTOL Air-Taxi Study**. 2016. Disponível em: <<http://www.nianet.org/ODM/ODMTuesdaypresentationsFinal/23MooreNASASiliconValleyStudy.pdf>>. Acesso em: June 1st 2019.

BARATA, J. C. A.; HUSSEIN, M. S. The Moore-Penrose Pseudoinverse: A Tutorial Review of the Theory. **Brazilian Journal of Physics**, v. 42, n. 1-2, p. 146–165, 2012. ISSN 01039733.

BENZAID, K.; MANSOURI, N. A Generalized Dynamical Model and Control Approach Applied to Multirotor Aerial Systems. In: **8th International Conference**

on Modelling, Identification and Control (ICMIC-2016). [S.l.: s.n.], 2016. ISBN 9780956715760.

BISSEL, C.; CHAPMAN, D. A. **Digital signal transmission.** 2nd. ed. Cambridge: Cambridge University Press, 1992. ISBN 0-521-41537-3.

BLAIN, L. **Russian unmanned, long-range, hybrid multirotor wants to do the heavy lifting.** 2017. Disponível em: <<https://newatlas.com/ardn-russia-skyf-heavy-lift-hybrid-multirotor/52300/>>. Acesso em: 27th February 2019.

BODSON, M. Evaluation of optimization methods for control allocation. **Journal of Guidance Control and Dynamics**, v. 25, p. 703–711, 2002.

BOEING. **Watch: Cargo Air Vehicle Completes First Outdoor Flight.** 2019. Disponível em: <<http://www.boeing.com/features/2019/05/cav-first-flight-05-19.page>>. Acesso em: June 1st 2019.

BORTOLIN, D. C. **Reguladores Robustos Recursivos para Sistemas Lineares sujeitos a Saltos Markovianos com Matrizes de Transição Incertas.** 2017. Tese (Doctorate) — University of São Paulo, 2017. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18153/tde-03092018-084228/pt-br.php>>.

BORTOLIN, D. C.; TERRA, M. H. Recursive robust regulator for Markovian jump linear systems subject to uncertain transition probabilities. In: **2015 European Control Conference (ECC).** IEEE, 2015. p. 3215–3219. ISBN 978-3-9524-2693-7. Disponível em: <<http://ieeexplore.ieee.org/document/7331029/>>.

BOYD, S.; HAST, M.; ÅSTRÖM, K. J. MIMO PID tuning via iterated LMI restriction. **International Journal of Robust and Nonlinear Control**, v. 26, n. 8, p. 1718–1731, may 2016. ISSN 10498923. Disponível em: <<http://doi.wiley.com/10.1002/rnc.3376>>.

BRESCIANINI, D.; D'ANDREA, R. Design, modeling and control of an omni-directional aerial vehicle. In: **2016 IEEE International Conference on Robotics and Automation (ICRA).** IEEE, 2016. p. 3261–3266. ISBN 978-1-4673-8026-3. ISSN 10504729. Disponível em: <<http://ieeexplore.ieee.org/document/7487497/>>.

BRITO, V. **Fault Tolerant Control of a X8-VB Quadcopter.** 2016. Tese (Master) — Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, 2016. Disponível em: <https://run.unl.pt/bitstream/10362/20249/1/Brito{_\}2016.>

BUFFINGTON, J. M.; ENNS, D. F. Lyapunov stability analysis of daisy chain control allocation. **Journal of Guidance Control and Dynamics**, v. 19, p. 1226–1230, 1996.

CAI, G. et al. A framework of frequency-domain flight dynamics modeling for multi-rotor aerial vehicles. **Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering**, v. 231, n. 1, p. 30–46, jan 2017. ISSN 0954-4100. Disponível em: <<http://pig.sagepub.com/lookup/doi/10.1177/0954410016648348http://journals.sagepub.com/doi/10.1177/0954410016648348>>.

CAMBONE, S. A. et al. **UAS Roadmap 2005-2030.** [S.l.], 2005. Disponível em: <https://fas.org/irp/program/collect/uav_roadmap2005.pdf>.

CASAVOLA, A.; GARONE, E. Fault-tolerant adaptive control allocation schemes for overactuated systems. **International Journal of Robust and Nonlinear Control**, v. 20, p. 1958–1980, 2010.

CERRI, J. P. **Regulador Robusto Recursivo para Sistemas Lineares de Tempo Discreto no Espaço de Estado**. 2009. Dissertação (Mestrado) — Universidade de São Paulo, 2009.

CERRI, J. P. **Controle e Filtragem para Sistemas Lineares Discretos Incertos sujeitos a Saltos Markovianos**. 2013. Tese (Doutorado) — Universidade de São Paulo, 2013.

CERRI, J. P.; TERRA, M. H. Recursive Robust Regulator for Discrete-Time Markovian Jump Linear Systems. **IEEE Transactions on Automatic Control**, v. 62, n. 11, p. 6004–6011, nov 2017. ISSN 0018-9286. Disponível em: <<http://ieeexplore.ieee.org/document/7932868/>>.

CHUNG, H. Y.; CHANG, S. H. Development of a combined algorithm of on-line instrument failure detection with an improved generalized likelihood ratio method and suboptimal control on a PWR pressurizer. **Nuclear Technology**, n. 74, p. 113–131, 1986.

COSTA, O. L. V. d.; MARQUES, R. P.; FRAGOSO, M. D. **Discrete-Time Markov Jump Linear Systems**. London: Springer, 2005. (Probability and Its Applications). ISSN 0018-9286. ISBN 978-1-85233-761-2. Disponível em: <<http://link.springer.com/10.1007/b138575>>.

CRAIG, J. J. **Introduction to Robotics**: Mechanics and control. 3rd. ed. Upper Saddle River, NJ 07458: Pearson, 2005.

DENATRAN. **Frota de Veículos - 2016**. 2019. Disponível em: <<http://www.denatran.gov.br/estatistica/639-frota-2019>>. Acesso em: June 1st 2019.

DIEBEL, J. **Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors**. Stanford University, 2006.

DOUCET, A.; FREITAS, N. de; GORDON, N. (Ed.). **Sequential Monte Carlo Methods in Practice**. New York: Springer Science, 2001. ISBN 978-1-4419-2887-0.

DURHAM, W. C. Constrained control allocation. **Journal of Guidance Control and Dynamics**, v. 16, p. 717–725, 1993.

FALCONI, G. P.; HEISE, C. D.; HOLZAPFEL, F. Fault-tolerant position tracking of a hexacopter using an Extended State Observer. In: **2015 6th International Conference on Automation, Robotics and Applications (ICARA)**. IEEE, 2015. v. 4435, n. 2, p. 550–556. ISBN 978-1-4799-6466-6. ISSN 1083-4435. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6525199http://ieeexplore.ieee.org/document/7081207/>>.

FALCONI, G. P.; HOLZAPFEL, F. Adaptive fault tolerant control allocation for a hexacopter system. In: **2016 American Control Conference (ACC)**. IEEE, 2016. p. 6760–6766. ISBN 978-1-4673-8682-1. ISSN 07431619. Disponível em: <<http://ieeexplore.ieee.org/document/7526736/>>.

FALCONI, G. P.; MARVAKOV, V. A.; HOLZAPFEL, F. Fault tolerant control for a hexarotor system using Incremental Backstepping. **2016 IEEE Conference on Control Applications, CCA 2016**, n. 1, p. 237–242, 2016.

FALCONI, G. P.; SCHATZ, S. P.; HOLZAPFEL, F. Fault tolerant control of a hexarotor using a command governor augmentation. **24th Mediterranean Conference on Control and Automation, MED 2016**, p. 182–187, 2016.

FECOMERCIO SP. **1h44 é o tempo médio gasto pelo paulistano no trânsito na cidade**. 2015. Disponível em: <<http://www.fecomercio.com.br/noticia/1h44-e-o-tempo-medio-gasto-pelo-paulistano-no-transito-na-cidade>>. Acesso em: Dec. 2nd 2016.

FRESK, E.; NIKOLAKOPOULOS, G. Full Quaternion Based Attitude Control for a Quadrotor. In: **European Control Conference**. [S.l.: s.n.], 2013. p. 3864–3869. ISBN 9783952417348. ISSN 14746670.

GIERAS, J. F. **Permanent Magnet Motor Technology**. Third. Boca Raton: CRC Press, 2010. 600 p. ISBN 9781420064407.

GIERNACKI, W.; SADALLA, T. Comparison of Tracking Performance and Robustness of Simplified Models of Multirotor UAV 's Propulsion Unit with CDM and PID Controllers (with anti- windup compensation) Comparison of Tracking Performance and Robustness of Simplified Models of Multirotor UAV 's Propulsion Unit with CDM and PID Controllers (with anti-windup compensation). n. September, p. 30–40, 2017.

GLOBO. **Embraer anuncia parceria com Uber para "carro voador"**. 2017. Disponível em: <<http://revistapegn.globo.com/Tecnologia/noticia/2017/04/embraer-anuncia-parceria-com-uber-para-carro-voador.html>>. Acesso em: 09 set. 2017.

GREEN CAR REPORTS. **1.2 Billion Vehicles On World's Roads Now, 2 Billion By 2035: Report**. 2014. Disponível em: <http://www.greencarreports.com/news/1093560_1-2-billion-vehicles-on-worlds-roads-now-2-billion-by-2035-report>. Acesso em: 02 dez. 2016.

GUANGXUN, D.; QUAN, Q.; KAI-YUAN, C. Additive-State-Decomposition-Based Dynamic Inversion Stabilized Control of A Hexacopter Subject to Unknown Propeller Damages. **Proceedings of the 32nd Chinese Control Conference**, p. 6231–6236, 2013. ISSN 21612927.

HAMILTON, W. R. On quaternions; or on a new system of imaginaries in algebra. **London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, v. 25, n. 3, p. 489–495, 1844.

HARKEGARD, O. Dynamic control allocation using constrained quadratic programming. **Journal of Guidance Control and Dynamics**, v. 27, p. 1028–1034, 2004.

HAWKINS, A. J. **Electric air taxi startup Lilium completes first test of its new five-seater aircraft**. 2019. Disponível em: <<https://www.theverge.com/2019/5/16/18625088/lilium-jet-test-flight-electric-aircraft-flying-car>>. Acesso em: June 1st 2019.

HERMANN, M.; SARAVI, M. **A First Course in Ordinary Differential Equations**. New Delhi: Springer India, 2014. 277 p. ISSN 0009-4978. ISBN 978-81-322-1834-0. Disponível em: <<http://link.springer.com/10.1007/978-81-322-1835-7>>.

- HESS, R. A.; WELLS, S. R. Sliding mode control applied to reconfigurable flight control design. **Journal of Guidance, Control, and Dynamics**, v. 26, n. 3, p. 452–462, 2003.
- HIBBEKER, R. C. **Engineering Mechanics: Dynamics**. 14. ed. Hoboken, New Jersey 07030: Pearson, 2016.
- ICHTEV, A. Multiple fuzzy models for fault tolerant control. **International Journal of Fuzzy Systems**, v. 5, n. 1, p. 31–40, 2003.
- INRIX. **Big or Small, All Cities Experience Rush Hour**. 2019. Disponível em: <<http://inrix.com/blog/2019/02/scorecard-2018/>>. Acesso em: June 1st 2019.
- IOANNOU, P. A.; SUN, J. **Robust Adaptive Control**. New Jersey: Dover, 2012. ISBN 978-0-486-49817-1.
- IPEA. **Acidentes de Transito nas Rodovias Federais - Caracterização, Tendencias e Custos para a Sociedade**. 2015.
- IRELAND, M.; VARGAS, A.; ANDERSON, D. A Comparison of Closed-Loop Performance of Multirotor Configurations Using Non-Linear Dynamic Inversion Control. **Aerospace**, v. 2, n. 2, p. 325–352, jun 2015. ISSN 2226-4310. Disponível em: <<http://www.mdpi.com/2226-4310/2/2/325/>>.
- ISERMANN, R.; SCHWARZ, R.; STOLZL, S. Fault-tolerant drive-by-wire systems. **IEEE Control Systems Magazine**, v. 22, n. 5, p. 64–81, 2002.
- JOHANSEN, T. A.; FOSSEN, T. I. Control allocation - A survey. **Automatica**, Elsevier Ltd, v. 49, n. 5, p. 1087–1103, may 2013. Disponível em: <<http://dx.doi.org/10.1016/j.automatica.2013.01.035><http://linkinghub.elsevier.com/retrieve/pii/S0005109813000368>>.
- KARASON, S. P.; ANNASWAMY, A. M. Adaptive Control in the Presence of Input Constraints. **IEEE Transactions on Automatic Control**, v. 39, n. 11, p. 2325–2330, 1994.
- KATZ, B. G. **Low Cost, High Performance Actuators for Dynamic Robots**. 2016. Tese (Undergraduate) — Massachusetts Institute of Technology, 2016. Disponível em: <<http://fab.cba.mit.edu/classes/865.18/motion/papers/katz-thesis.pdf>>.
- KELLEHER, S. R. **Pilotless Electric Flying Taxis Are Coming To Singapore**. 2019. Disponível em: <<https://www.forbes.com/sites/suzannerowakelleher/2019/05/27/pilotless-electric-flying-taxis-are-coming-to-singapore/#3c027b096619>>. Acesso em: June 1st 2019.
- LAWRENCE, J. D. **A Catalog of Special Plane Curves**. New York: Dover Publications, Inc., 1972. 218 p. ISBN 0-486-60288-5.
- LEÃO, W.; INOQUE, R.; TERRA, M. Análise Comparativa de Controles Robustos Aplicados em um Quadricóptero. **XII Simpósio Brasileiro de Automação Inteligente (SBAI)**, p. 1–6, 2015.
- LEAO, W. M. **Análise Comparativa de Controladores Robustos Aplicados em Robôs Móvel e Aéreo**. 2015. Dissertação (Master Thesis) — São Carlos School of Engineering, University of São Paulo, 2015.

LEE, T.; LEOK, M.; MCCLAMROCH, N. H. Control of Complex Maneuvers for a Quadrotor UAV using Geometric Methods on SE(3). **CoRR**, abs/1003.2, n. March, mar 2010. Disponível em: <<http://arxiv.org/abs/1003.2005>>.

LIAO, F.; WANG, J. L.; YANG, G.-H. Reliable robust flight tracking control: An LMI approach. **IEEE Transactions on Control Systems Technology**, v. 10, n. 1, p. 76–89, 2003.

LIU, H. et al. Robust optimal attitude control of multirotors. **Australasian Conference on Robotics and Automation, ACRA**, p. 2–4, 2013. ISSN 14482053.

LOZANO, S. **First Steps Toward Drone Traffic Management**. 2015. Disponível em: <<https://www.nasa.gov/feature/ames/first-steps-toward-drone-traffic-management>>. Acesso em: June 1st 2019.

LUUKKONEN, T. Modelling and Control of Quadcopter. 2011.

MAHONY, R.; KUMAR, V.; CORKE, P. Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor. **IEEE Robotics & Automation Magazine**, v. 19, n. 3, p. 20–32, sep 2012. ISSN 1070-9932. Disponível em: <<http://ieeexplore.ieee.org/document/6289431/>>.

MARKS, A.; WHIDBORNE, J. F.; YAMAMOTO, I. Control allocation for fault tolerant control of a vtol octorotor. In: **Proceedings of 2012 UKACC International Conference on Control**. IEEE, 2012. p. 357–362. ISBN 978-1-4673-1560-9. Disponível em: <<http://ieeexplore.ieee.org/document/6334656/>>.

MATHWORKS. **ga: find minimum of function using genetic algorithm**. 2019. Disponível em: <<https://www.mathworks.com/help/gads/ga.html>>. Acesso em: 13th may 2019.

_____. **Genetic Algorithm Options**. 2019. Disponível em: <<https://www.mathworks.com/help/gads/genetic-algorithm-options.html#f7820>>. Acesso em: 13th may 2019.

_____. **ode15s**. 2019. Disponível em: <<https://www.mathworks.com/help/matlab/ref/ode15s.html>>. Acesso em: 27th april 2019.

_____. **ode45**. 2019. Disponível em: <<https://www.mathworks.com/help/matlab/ref/ode45.html>>. Acesso em: 27th april 2019.

_____. **Ordinary Differential Equations**. 2019. Disponível em: <<https://www.mathworks.com/help/matlab/ordinary-differential-equations.html>>. Acesso em: 27th april 2019.

MERHEB, A.-R.; BATEMAN, F.; NOURA, H. Passive and active fault tolerant control of octorotor UAV using Second Order Sliding Mode control. In: **2015 IEEE Conference on Control Applications (CCA)**. [s.n.], 2015. p. 1907–1912. ISBN 978-1-4799-7787-1. ISSN 1085-1992. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7320888>>.

MICHIELETTI, G.; RYLL, M.; FRANCHI, A. Control of statically hoverable multi-rotor aerial vehicles and application to rotor-failure robustness for hexarotors. In: **2017 IEEE International Conference on Robotics and Automation (ICRA)**.

- IEEE, 2017. p. 2747–2752. ISBN 978-1-5090-4633-1. ISSN 10504729. Disponível em: <<http://ieeexplore.ieee.org/document/7989320/>>.
- MOHAMMADI, M.; Mohammad Shahri, A. Modelling and decentralized adaptive tracking control of a quadrotor UAV. In: **2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)**. IEEE, 2013. p. 293–300. ISBN 978-1-4673-5811-8. Disponível em: <http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6510122http://ieeexplore.ieee.org/document/65101>.
- MONTOYA, R. J. et al. **Restructurable controls**. Tech. Rep. NASA CP-2277. NASA Langley Research Center, Hampton, VA, USA, 1983.
- NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION. **Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey**. 2015. Disponível em: <<https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115>>. Acesso em: June 1st 2019.
- NAVABI, M.; MIRZAEI, H. Robust Optimal Adaptive Trajectory Tracking Control of Quadrotor Helicopter. **Latin American Journal of Solids and Structures**, v. 14, n. 6, p. 1040–1063, jun 2017. ISSN 1679-7825.
- NGUYEN, N. P.; HONG, S. K. Fault-tolerant Control of Quadcopter UAVs Using Robust Adaptive Sliding Mode Approach. **Energies**, v. 12, n. 1, p. 95, dec 2018. ISSN 1996-1073. Disponível em: <<http://www.mdpi.com/1996-1073/12/1/95>>.
- OPPENHEIMER, M.; DOMAN, D.; BOLENDER, M. Control allocation. In: LEVINE, W. S. (Ed.). **The control handbook: control system applications**. 2nd. ed. [S.l.]: CRC Press, 2010. cap. 8.
- ÖZBEK, N. S.; ÖNKOL, M.; EFE, M. Ö. Feedback control strategies for quadrotor-type aerial robots: a survey. **Transactions of the Institute of Measurement and Control**, v. 38, n. 5, p. 529–554, may 2016. ISSN 0142-3312. Disponível em: <<http://journals.sagepub.com/doi/10.1177/0142331215608427>>.
- PANOMRATTANARUG, B.; HIGUCHI, K.; MORA-CAMINO, F. Attitude control of a quadrotor aircraft using LQR state feedback controller with full order state observer. In: **Proceedings of the SICE Annual Conference**. [S.l.: s.n.], 2013. p. 2041–2046.
- POUNDS, P. et al. Towards Dynamically-Favourable Quad-Rotor Aerial Robots. In: **Australasian Conference on Robotics and Automation**. [S.l.: s.n.], 2004. p. 10. ISBN 0-9587583-6-0.
- PRAKASH, J.; PATWARDHAN, S. C.; NARASIMHAN, S. A supervisory approach to fault-tolerant control of linear multivariable systems. **Industrial and Engineering Chemistry Research**, v. 41, n. 9, p. 2270–2281, 2002.
- QI, X. et al. Fault diagnosis and fault tolerant control methods for manned and unmanned helicopters: A literature review. **Conference on Control and Fault-Tolerant Systems, SysTol**, p. 132–139, 2013. ISSN 21621195.
- RINALDI, F.; GARGIOLI, A.; QUAGLIOTTI, F. PID and LQ Regulation of a Multirotor Attitude: Mathematical Modelling, Simulations and Experimental Results. **Journal of Intelligent & Robotic Systems**, v. 73, n. 1-4, p. 33–50, jan 2014. ISSN 0921-0296. Disponível em: <<http://link.springer.com/10.1007/s10846-013-9911-x>>.

RUFF, R.; STEPHENS, C.; MAHAPATRA, S. Applying Model-Based Design to Large-Scale Systems Development: Modeling, Simulation, Test, & Deployment of a Multirotor Vehicle. **AIAA Modeling and Simulation**, p. 1–17, 2012. Disponível em: <<http://arc.aiaa.org/doi/pdf/10.2514/6.2012-4939>>.

SAIED, M. et al. Fault Diagnosis and Fault-Tolerant Control Strategy for Rotor Failure in an Octorotor. **IEEE International Conference on Robotics and Automation (ICRA)**, p. 5266–5271, 2015.

_____. Passive fault-tolerant control of an octorotor using super-twisting algorithm: Theory and experiments. In: **2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)**. IEEE, 2016. p. 361–366. ISBN 978-1-5090-0658-8. ISSN 21621209. Disponível em: <<http://ieeexplore.ieee.org/document/7739777/>>.

_____. Passive fault-tolerant control of an octorotor using super-twisting algorithm: Theory and experiments. In: **2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)**. IEEE, 2016. p. 361–366. ISBN 978-1-5090-0658-8. ISSN 21621209. Disponível em: <<http://ieeexplore.ieee.org/document/7739777/>>.

SALTELLI, A. et al. **Sensitivity analysis in practice: a guide to assessing scientific models**. West Sussex: John Wiley & Sons, Ltd., 2004. 219 p. ISBN 0-470-87093-1. Disponível em: <<http://books.google.com/books?id=NsAVmohPNpQC&pgi>>.

SANTOS, M. F. et al. Active Fault-Tolerant Control applied to a hexacopter under propulsion system failures. In: **2015 19th International Conference on System Theory, Control and Computing, ICSTCC 2015 - Joint Conference SINTES 19, SACCS 15, SIMSIS 19**. [S.l.: s.n.], 2015. p. 447–453. ISBN 9781479984817.

SATICI, a. C.; POONAWALA, H.; SPONG, M. W. Robust Optimal Control of Quadrotor UAVs. **Access, IEEE**, v. 1, p. 79–93, 2013. ISSN 2169-3536.

SCHWAGER, M.; ANNASWAMY, A. M.; LAVRETSKY, E. Adaptation-Based Reconfiguration in the Presence of Actuator Failures and Saturation. **Proc. of the American Control Conference**, p. 2640–2645, 2005. ISSN 07431619.

SHI, J. et al. Research on allocation efficiency of the redistributed pseudo inverse algorithm. **Science China Information Sciences**, v. 53, p. 271–277, 2010.

SHRAIM, H.; AWADA, A.; YOUNESS, R. A survey on quadrotors: Configurations, modeling and identification, control, collision avoidance, fault diagnosis and tolerant control. **IEEE Aerospace and Electronic Systems Magazine**, IEEE, v. 33, n. 7, p. 14–33, jul 2018. ISSN 0885-8985. Disponível em: <<https://ieeexplore.ieee.org/document/8408892/>>.

SIQUEIRA, A. A. G.; TERRA, M. H.; BERGERMAN, M. **Robust Control of Robots**. London: Springer, 2011. ISBN 978-0-85729-897-3. Disponível em: <<http://link.springer.com/10.1007/978-0-85729-898-0>>.

SPURGEON, S. Sliding mode control : a tutorial. **Proc. of the European Control Conference (ECC), 2014**, n. 1, p. 2272–2277, 2014.

TIMES OF INDIA. **Is commute time taking over your life?** 2008. Disponível em: <<http://timesofindia.indiatimes.com/life-style/Is-commute-time-taking-over-your-life/articleshow/3508745.cms>>. Acesso em: 02 dez. 2016.

UBER ELEVATION. **Fast-Forwarding to a Future of On-Demand Urban Air Transportation.** 2016.

UNITED STATES CENSUS BUREAU. **Average One-Way Commuting Time by Metropolitan Areas.** 2016. Disponível em: <<https://www.census.gov/library/visualizations/interactive/travel-time.html>>. Acesso em: June 1st 2019.

VEILLETTE, R. J. Reliable linear-quadratic state-feedback control. **IEEE Control Systems Magazine**, v. 31, n. 1, p. 137–143, 1995.

WEISS, R.; NICOLA, S. **Flying Taxi Startups Target Asia Debut Using European Technology.** 2019. Disponível em: <<https://www.bloomberg.com/news/articles/2019-05-23/china-air-taxis-are-next-frontier-for-austrian-aviation-supplier>>. Acesso em: June 1st 2019.

WORLD HEALTH ORGANIZATION. **Global status report on road safety 2018.** 2018. Disponível em: <https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/>.

XIONG, J.-J.; ZHANG, G.-B. Global fast dynamic terminal sliding mode control for a quadrotor UAV. **ISA Transactions**, v. 66, p. 233–240, jan 2017. ISSN 00190578. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0019057816303627>>.

YEN, G. G. Reconfigurable learning control in large space structures. **IEEE Transactions on Control System Technology**, v. 2, n. 4, p. 362–370, 1994.

YOON, H.-J.; CICHELLA, V.; HOVAKIMYAN, N. Robust Adaptive Control Allocation for an Octocopter under Actuator Faults. In: **AIAA Guidance, Navigation, and Control Conference**. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2016. p. 1–16. ISBN 978-1-62410-389-6. Disponível em: <<http://arc.aiaa.org/doi/10.2514/6.2016-0635>>.

ZHANG, Y.; JIANG, J. Bibliographical review on reconfigurable fault-tolerant control systems. **Annual Reviews in Control**, v. 32, n. 2, p. 229–252, 2008. ISSN 13675788.

Appendix

APPENDIX A – LARGE-SIZE COAXIAL OCTACOPTER CONSTRUCTION

Figures 48 to 53 present a few steps of the drone construction process until its flight test using standard factory control algorithms.



Figure 48: Drone parts.



Figure 49: Assembling (1)

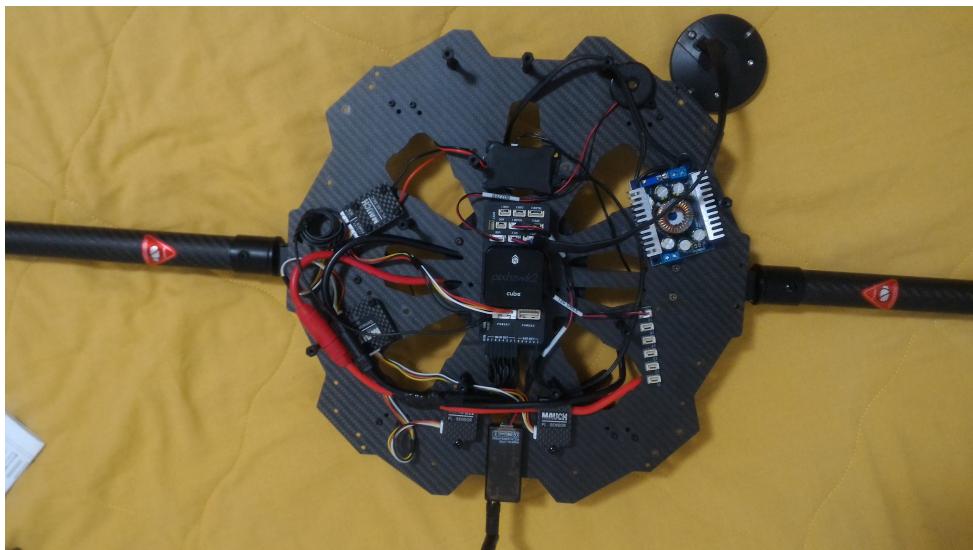


Figure 50: Assembling (2)



Figure 51: Assembling (3)



Figure 52: Drone completed



Figure 53: Flight test

APPENDIX B – MULTICOPTER SIMULATION SCRIPT EXAMPLE

```

1 % Octorotor simulation
2 % refer to "doc multicontrol" or "doc multicopter" for more ...
   information
3
4 addpath('..../multiControl/')
5 addpath('..../multiControl/utils')
6 warning('off','all')
7 clear all
8 clc
9 %% Configuration for +8 coaxial octorotor
10 % Creates simulation class
11 multirotor = multicontrol(8);
12 % multirotor.supressVerbose()
13 % Define rotor positions
14 positions = [[0.34374 0.34245 0.0143]', ...
15                 [-0.341 0.34213 0.0143]', ...
16                 [-0.34068 -0.34262 0.0143]', ...
17                 [0.34407 -0.34229 0.0143]', ...
18                 [0.33898 0.33769 0.0913]', ...
19                 [-0.33624 0.33736 0.0913]', ...
20                 [-0.33591 -0.33785 0.0913]', ...
21                 [0.3393 -0.33753 0.0913']];
22 multirotor.setRotorPosition(1:8,positions);
23 % Define rotor orientations
24 orientations = [[-0.061628417 -0.061628417 0.996194698]', ...
25                         [0.061628417 -0.061628417 0.996194698]', ...
26                         [0.061628417 0.061628417 0.996194698]', ...
27                         [-0.061628417 0.061628417 0.996194698]', ...
28                         [-0.061628417 -0.061628417 0.996194698]', ...
29                         [0.061628417 -0.061628417 0.996194698]', ...
30                         [0.061628417 0.061628417 0.996194698]', ...
31                         [-0.061628417 0.061628417 0.996194698']];
32 multirotor.setRotorOrientation(1:8,orientations);
33 % Define aircraft's inertia
34 multirotor.setMass(6.015);
35 mass = 3;
36 inertia = [0.3143978800, 0.0000861200, -0.0014397600

```

```
37          0.0000861200, 0.3122127800, 0.0002368800  
38          -0.0014397600, 0.0002368800, 0.5557912400];  
39 multirotor.setInertia(inertia);  
40 payloadRadius = 0.3*mean(sqrt(sum(positions.^2)));  
41 multirotor.setPayload([0, 0, ...  
    -payloadRadius],mass,eye(3)*2*mass*payloadRadius*payloadRadius/5);  
42 % Define aircraft's drag coeff  
43 friction = [0.25, 0, 0  
    0, 0.25, 0  
    0, 0, 0.25];  
44 multirotor.setFriction(friction);  
45 multirotor.setAngularFilterGain([0,0,0.9]);  
46 % Define lift and drag coefficients  
47 speed = [0  
    200  
    416.5751859  
    435.2676622  
    462.5052705  
    472.6526147  
    491.345091  
    501.4924353  
    520.1849116  
    530.3322559  
    549.0247321  
    567.7172084  
    586.4096847  
    748.2865294  
    1000];  
48  
49 liftCoeff = [0.00004  
    0.00007  
    0.00009663400821486720  
    0.00010197039400480800  
    0.00010177480503994200  
    0.00010886498777293000  
    0.00011048831185009000  
    0.00011230119869840700  
    0.00010908666646728400  
    0.00011227432775784800  
    0.00010996476733082600  
    0.00010862374599149600  
    0.00010409054272222600  
    0.00006567742093581670
```

```

78          0];
79 dragCoeff = [0.0000005
80          0.00000075
81          0.00000115158401406177
82          0.00000131849846466781
83          0.00000140132963964922
84          0.00000156543968817590
85          0.00000165553807692624
86          0.00000178787094426600
87          0.00000184631980295481
88          0.00000195397512083756
89          0.00000198893164777812
90          0.00000201512348657737
91          0.00000203398711313428
92          0.00000136514255905061
93          0];
94 multirotor.setRotorLiftCoeff(1:8,[speed ...
95          liftCoeff], 'smoothingspline');
95 multirotor.setRotorDragCoeff(1:8,[speed ...
96          dragCoeff], 'smoothingspline');
96 % Define rotor inertia
97 multirotor.setRotorInertia(1:8,0.00047935*ones(1,8));
98 % Sets rotor parameters
99 rotationDirection = [1 -1 1 -1 -1 1 -1 1]';
100 multirotor.setRotorDirection(1:8,rotationDirection);
101 multirotor.setRotorMaxSpeed(1:8,729*ones(1,8));
102 multirotor.setRotorMinSpeed(1:8,0*ones(1,8));
103 multirotor.setInitialRotorSpeeds(328*rotationDirection);
104 multirotor.setInitialInput(9.47*rotationDirection);
105 multirotor.setRotorRm(1:8,0.0975*ones(1,8));
106 multirotor.setRotorKt(1:8,0.02498*ones(1,8));
107 multirotor.setRotorKv(1:8,340*ones(1,8));
108 multirotor.setRotorMaxVoltage(1:8,22*ones(1,8));
109 multirotor.setRotorOperatingPoint(1:8,352*[1 1 1 1 1 1 1 1]);
110 %% Controller configuration
111 % Trajectory controller
112 % PID:
113 kp = [95.75 100.250020593405 150.0000495910645];
114 ki = [14.0625572204590 13 19];
115 kd = [49.5000000000000 49.2500000000000 60];
116 kdd = [11.5001522749662 7.2500000000000 13.2500582933426];
117 % PID attitude controller

```

```

118 kp = [133.015695810318 139.750023871660 100.0000053495169];
119 ki = [205.937575563788 206 202];
120 kd = [23.0000121444464 19.2501215338707 7];
121 multirotor.configController('PID',kp,ki,kd);
122 % Control allocation
123 multirotor.configControlAllocator('Passive NMAC',1,0);
124 %% Simulator configuration
125 multirotor.setTimeStep(0.005);
126 multirotor.setControlTimeStep(0.05);
127 multirotor.setController('PID');
128 multirotor.setControlAllocator('Active NMAC');
129 multirotor.setAttitudeReferenceCA('Active NMAC');
130 multirotor.configFDD(.98,0.25)
131 endTime = 15;
132 [waypoints, time] = geronoToWaypoints(7, 4, ...
    4, endTime, endTime/8, 'goto', 2*pi);
133 multirotor.setTrajectory('waypoints',waypoints,time);
134 multirotor.addCommand({'setRotorStatus(1,''motor ...
    loss'',0.001)'},endTime/2)
135 multirotor.addCommand({'setRotorStatus(3,''motor ...
    loss'',0.001)'},0)
136 multirotor.setSimEffects('motor dynamics on','solver euler')
137 multirotor.setLinearDisturbance('@(t) ...
    [0;1;0]*10*exp(-(t-3.75)^2/(0.5))')
138 multirotor.setControlDelay(0.20);
139 %% Run simulation
140 multirotor.run('visualizeGraph',false,'visualizeProgress',true,'metricPrecision'
141 multirotor.plotSim();

```

APPENDIX C – MODEL 2 MOTOR CHARACTERIZATION TABLE

ML5208								
Item NO.	Volts (V)	Prop	Throttle	Amps (A)	Watts (W)	Thrust (g)	Efficiency (g/W)	Operating temperature(°C)
ML5208 340KV	24V	1555	50%	3.7	88.8	910	10.25	42°C
			54%	5	120.0	1120	9.33	
			57%	6	144.0	1290	8.96	
			62%	8	192.0	1540	8.02	
			66%	10	240.0	1820	7.58	
			70%	12	288.0	2070	7.19	
			74%	14	336.0	2280	6.79	
			78%	16	384.0	2470	6.43	
			81%	18	432.0	2630	6.09	
		1755	100%	20	480.0	2880	6.00	65°C
			50%	5.3	127.2	1220	9.59	
			52%	6	144.0	1320	9.17	
			56%	8	192.0	1620	8.44	
			59%	10	240.0	1900	7.92	
			62%	12	288.0	2140	7.43	
			65%	14	336.0	2340	6.96	
			68%	16	384.0	2610	6.80	
			71%	18	432.0	2750	6.37	
		1855	75%	20	480.0	2930	6.10	78°C
			78%	22	528.0	3120	5.91	
			80%	24	576.0	3290	5.71	
			82%	26	624.0	3440	5.51	
			100%	27.8	667.2	3560	5.34	
			50%	6.6	158.4	1480	9.34	
			52%	8	192.0	1710	8.91	
			55%	10	240.0	1970	8.21	
			59%	12	288.0	2220	7.71	
			61%	14	336.0	2480	7.38	
			64%	16	384.0	2720	7.08	
			66%	18	432.0	2880	6.67	
			69%	20	480.0	3010	6.27	
			71%	22	528.0	3220	6.10	
			74%	24	576.0	3380	5.87	
			77%	26	624.0	3570	5.72	
			80%	28	672.0	3650	5.43	
			100%	30.6	734.4	3750	5.11	

Notes: The test condition of temperature is motor surface temperature in 100% throttle while the motor run 10 min. environment temperature 26°C

Figure 54: Manufacturer characterization of motor for Model 2 construction and simulation.

APPENDIX D – PSEUDO-CODES

Algorithm 3 Desired trajectory calculation method from Section 3.1

```

1: function DESIREDTRAJECTORY(time)
2:    $t_0, t_f, \mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z, \mathbf{p}_{\hat{\psi}} \leftarrow$  choose which waypoint to follow considering the current
   time and previously stored waypoints from user interface
3:    $\Delta t \leftarrow$  time -  $t_0$ 
4:   for  $i = \hat{x}, \hat{y}, \hat{z}, \hat{\psi}$  do
5:      $\mathbf{a}_i \leftarrow$  Equation (3.2) for  $t_0, t_f, \mathbf{p}_i$ 
6:      $p_i^d, \dot{p}_i^d, \ddot{p}_i^d \leftarrow$  Equation (3.1) for  $\Delta t, \mathbf{a}_i$ 
7:   end for
8:   desiredState  $\leftarrow p_i^d, \dot{p}_i^d, \ddot{p}_i^d$  for  $i = \hat{x}, \hat{y}, \hat{z}, \hat{\psi}$ 
9:   return desiredState
10: end function
```

Algorithm 4 Fault Detection and Diagnose simulation from Section 4.1.6

```

1: function FDD(time)
2:   buffer  $\leftarrow$  read rotor statuses from memory
3:   for  $i = 1, \dots, N$  do
4:     buffer $_i(end + 1) \leftarrow \xi_i$ 
5:     for  $j=1, \dots, \text{LENGTH}(\text{buffer}_i)$  do
6:       if time of buffer $_i(j) <$  time - fddDelay then
7:         remove buffer $_i(j)$ 
8:       end if
9:     end for
10:    error  $\leftarrow (1 - \text{fddHitRate}) * \text{RANDN}()$             $\triangleright \text{RANDN}()$  returns a normally
      distributed pseudorandom number
11:    test1  $\leftarrow$  buffer $_i(1) + \text{error}$ 
12:    if test1  $> 1$  then
13:      test1  $\leftarrow 1$ 
14:    else if test1  $\leq 0$  then
15:      test1  $\leftarrow 0$ 
16:    end if
17:    test2  $\leftarrow$  buffer $_i(1) - \text{error}$ 
18:    if test2  $> 1$  then
19:      test2  $\leftarrow 1$ 
20:    else if test2  $\leq 0$  then
21:      test2  $\leftarrow 0$ 
```

```

22:    | end if
23:    | if  $|\text{buffer}_i(1) - \text{test1}| \geq |\text{buffer}_i(1) - \text{test2}|$  then
24:    |   |  $\xi_{i,fdd} \leftarrow \text{test1}$ 
25:    | else
26:    |   |  $\xi_{i,fdd} \leftarrow \text{test2}$ 
27:    | end if
28:  end for
29:  diagnosis  $\leftarrow \xi_{i,fdd}, \forall i = 1, \dots, N$ 
30:  return diagnosis
31: end function

```

Algorithm 5 Position controller from Section 3.2

```

1: function POSITIONCONTROL(desState, currState)
2:    $p_i^d, \dot{p}_i^d, \ddot{p}_i^d$  for  $i = \hat{x}, \hat{y}, \hat{z}, \hat{\psi} \leftarrow \text{desState}$ 
3:    $P_{Ai}^d, v_{Ai}^d, \dot{v}_{Ai}^d$  for  $i = \hat{x}, \hat{y}, \hat{z}, \hat{\psi} \leftarrow \text{currState}$ 
4:   posCtrl  $\leftarrow$  Equation (3.3)
5:   return posCtrl
6: end function

```

Algorithm 6 Attitude planning method of Section 3.3

```

1: function ATTITUDEREFERENCE(desImpulse, desYaw, diagnosis)
2:    $\psi \leftarrow \text{desYaw}$ 
3:    $q_\psi \leftarrow$  Equation (3.5)
4:    $Q_{ca} \leftarrow$  Equation (3.6)
5:    $\mathbf{T}_C^d \leftarrow Q_{ca}^{-1}\text{desImpulse}$ 
6:    $M_f \leftarrow$  Equation (3.8)
7:    $M_t \leftarrow$  Equation (3.43)
8:   attPlanningType,  $\omega^{min}, \omega^{max}, W_m, W_a \leftarrow$  read values stored from previous user
   interface
9:    $\xi_{i,fdd}, \forall i = 1, \dots, N \leftarrow \text{diagnosis}$ 
10:  switch attPlanningType do
11:    | case PI Passive
12:    |   |  $\Omega' \leftarrow M_f^\dagger \mathbf{T}_C^d$ 
13:    |   |  $\Omega'(\Omega' < 0) \leftarrow 0$ 
14:    | end case
15:    | case PI Active
16:    |   |  $\Lambda_{fdd} \leftarrow$  Equations (3.45) and (4.10)
17:    |   |  $\Omega' \leftarrow (M_f \Lambda_{fdd})^\dagger \mathbf{T}_C^d$ 
18:    |   |  $\Omega'(\Omega' < 0) \leftarrow 0$ 
19:    | end case

```

```

20:   | case RPI Passive           ▷ RPI stands for Redistributed Pseudo-Inverse
21:   |   |  $\Omega' \leftarrow$  Algorithm 1 with  $H \leftarrow M_f$ ,  $Q^{-1}\mathbf{T}_A^d \leftarrow \mathbf{T}_C^d$  and  $\Gamma_B^d \leftarrow []$ 
22:   | end case
23:   | case RPI Active
24:   |   |  $\Lambda_{fdd} \leftarrow$  Equations (3.45) and (4.10)
25:   |   |  $\Omega' \leftarrow$  Algorithm 1 with  $H \leftarrow M_f\Lambda_{fdd}$ ,  $Q^{-1}\mathbf{T}_A^d \leftarrow \mathbf{T}_C^d$ ,  $\Gamma_B^d \leftarrow []$  and the
      relation of Equations (3.125) and (3.126)
26:   | end case
27:   | case NSCA Passive ▷ NSCA stands for Null-Space based Control Allocation
28:   |   |  $\Omega_{1/2} \leftarrow$  Equation (??)
29:   |   |  $N_s \leftarrow \text{NULL}(M_t)$  ▷  $\text{NULL}(M_t)$  returns the null space base vectors of  $M_t$ 
30:   |   |  $\mathbf{T}_B^d \leftarrow \mathbf{T}_C^d$ 
31:   |   |  $\Omega \leftarrow$  Equation (3.131)
32:   |   |  $\Omega' \leftarrow$  Equation (3.10)
33:   | end case
34:   | case NSCA Active
35:   |   |  $\Lambda_{fdd} \leftarrow$  Equations (3.45) and (4.10)
36:   |   |  $\omega^{min}, \omega^{max} \leftarrow$  Equations (3.125) and (3.126)
37:   |   |  $\Omega_{1/2} \leftarrow$  Equation (??)
38:   |   |  $N_s \leftarrow \text{NULL}(M_t\Lambda_{fdd})$ 
39:   |   |  $\mathbf{T}_B^d \leftarrow \mathbf{T}_C^d$ 
40:   |   |  $\Omega \leftarrow$  Equation (3.131)
41:   |   |  $\Omega' \leftarrow$  Equation (3.10)
42:   | end case
43: end switch
44:  $\mathbf{T}_A \leftarrow$  Equation (3.11)
45: if desImpulse $_{\hat{z}} < 0$  then
46:   | desImpulse $_{\hat{z}} \leftarrow -\text{desImpulse}_{\hat{z}}$ 
47: end if
48: if  $|\mathbf{T}_A| = 0$  then
49:   |  $\mathbf{T}_A \leftarrow [0, 0, 1]^T$ 
50: end if
51:  $\mathbf{T}_A^d \leftarrow \text{desImpulse}$ 
52:  $q_{cb} \leftarrow$  Equation (3.12)
53: desAtt  $\leftarrow$  Equation (3.16)
54: return desAtt
55: end function

```

```

1: function ATTITUDECONTROL(desAtt, posCtrl, diagnosis)
2:    $q_{ref} \leftarrow desAtt$ 
3:    $\mathbf{q}_e \leftarrow$  Equation (3.17)
4:    $\mathbf{q}_e \leftarrow$  Equation (3.18)
5:    $\dot{\omega}_B^d \leftarrow$  Equation (4.2)
6:    $M_f \leftarrow$  Equation (3.8)
7:    $M_t \leftarrow$  Equation (3.43)
8:   attControllerType,  $\leftarrow$  read values stored from previous user interface
9:    $\xi_{i,fdd}, \forall i = 1, \dots, N \leftarrow$  diagnosis
10:   $\mathbf{T}_A^d \leftarrow posCtrl$ 
11:  switch attControllerType do
12:    case PID (Section 3.4.1)
13:       $\Gamma_B^d \leftarrow$  Equation (3.20)
14:    end case
15:    case SOSMC Passive (Section 3.4.2.1)
16:       $a(t) \leftarrow$  Equation (2.25)
17:       $f(x), g(x), \ddot{x}, \ddot{x}_d, \dot{e}, e \leftarrow$  Equation (3.34)
18:       $s \leftarrow$  Equation (3.22)
19:       $\Gamma_B^d \leftarrow$  Equation (3.25)
20:      attCtrl  $\leftarrow \Gamma_B^d$ 
21:    end case
22:    case SOSMC Active (Section 3.4.2.1)
23:       $a_{fdd}(t) \leftarrow$  Equations (3.35) and (4.9)
24:       $f(x), g(x), \ddot{x}, \ddot{x}_d, \dot{e}, e \leftarrow$  Equation (3.36)
25:       $s \leftarrow$  Equation (3.22)
26:       $\Gamma_B^d \leftarrow$  Equation (3.25)
27:      attCtrl  $\leftarrow \Gamma_B^d$ 
28:    end case
29:    case SOSMC Passive with PIDD (Section 3.4.2.3)
30:       $\mathbf{T}_A^d \leftarrow posCtrl + \mathbf{g}_A$ 
31:       $\mathbf{e}_{P_B} \leftarrow$  Equation (3.38)
32:       $a(t) \leftarrow$  Equation (2.25)
33:       $f(x), g(x), \ddot{x}, \ddot{x}_d, \dot{e}, e \leftarrow$  Equation (3.39)
34:       $s \leftarrow$  Equation (3.22)
35:       $\Gamma_B^d, \mathbf{T}_A^{d*} \leftarrow$  Equation (3.25)
36:      attCtrl  $\leftarrow \Gamma_B^d, \mathbf{T}_A^{d*}$ 
37:    end case
38:    case SOSMC Active with PIDD (Section 3.4.2.4)
39:       $\mathbf{T}_A^d \leftarrow posCtrl + \mathbf{g}_A$ 

```

```

40:    $\mathbf{e}_{\mathbf{P}_B} \leftarrow$  Equation (3.38)
41:    $a_{fdd}(t) \leftarrow$  Equations (3.35) and (4.9)
42:    $f(x), g(x), \dot{x}, \dot{x}_d, \dot{e}, e \leftarrow$  Equation (3.40)
43:    $s \leftarrow$  Equation (3.22)
44:    $\boldsymbol{\Gamma}_B^d, \mathbf{T}_A^{d*} \leftarrow$  Equation (3.25)
45:   attCtrl  $\leftarrow \boldsymbol{\Gamma}_B^d, \mathbf{T}_A^{d*}$ 
46: end case
47: case SOSMC Passive Direct (Section 3.4.2.5)
48:    $\mathbf{T}_A^d \leftarrow$  posCtrl +  $\mathbf{g}_A$ 
49:    $a(t) \leftarrow$  Equation (2.25)
50:    $f(x), \ddot{x}, \ddot{x}_d, \dot{e}, e \leftarrow$  Equation (3.44)
51:    $\boldsymbol{\Omega}_{1/2} \leftarrow$  Equation (??)
52:    $N_s \leftarrow \text{NULL}(M_t)$ 
53:   accelerations  $\leftarrow (\ddot{x}^d + c\dot{e} - f(x))$ 
54:    $\boldsymbol{\Gamma}_B^d \leftarrow I_{cg}$  accelerations1:3
55:    $\mathbf{T}_B^d \leftarrow m$  accelerations4:6
56:    $\mathbf{b}_2 \leftarrow$  Equation (3.131)
57:    $c \leftarrow$  Equation (3.134)
58:    $\boldsymbol{\Omega}_f \leftarrow$  Equation (3.128)
59:    $\boldsymbol{\Omega}_\tau \leftarrow$  Equation (3.129)
60:    $u_{eq} \leftarrow \boldsymbol{\Omega}_\tau + \boldsymbol{\Omega}_f$ 
61:    $s \leftarrow$  Equation (3.22)
62:    $u_{dis} \leftarrow$  Equation (3.27)
63:    $\boldsymbol{\Omega}^d \leftarrow$  Equation (3.25)
64:    $\boldsymbol{\omega}^d \leftarrow$  Equation (3.121)
65:   attCtrl  $\leftarrow \boldsymbol{\omega}^d$ 
66: end case
67: case SOSMC Active Direct (Section 3.4.2.6)
68:    $\mathbf{T}_A^d \leftarrow$  posCtrl +  $\mathbf{g}_A$ 
69:    $\Lambda_{fdd} \leftarrow$  Equations (3.45) and (4.10)
70:    $a_{fdd}(t) \leftarrow$  Equations (3.35) and (4.9)
71:    $f(x), \ddot{x}, \ddot{x}_d, \dot{e}, e \leftarrow$  Equation (3.46)
72:    $M_f, M_t, \boldsymbol{\omega}^{min}, \boldsymbol{\omega}^{max} \leftarrow$  Equations (3.137), (3.138), (3.139) and (3.140)
73:    $\boldsymbol{\Omega}_{1/2} \leftarrow$  Equation (??)
74:    $N_s \leftarrow \text{NULL}(M_t)$ 
75:   accelerations  $\leftarrow (\ddot{x}^d + c\dot{e} - f(x))$ 
76:    $\boldsymbol{\Gamma}_B^d \leftarrow I_{cg}$  accelerations1:3
77:    $\mathbf{T}_B^d \leftarrow m$  accelerations4:6
78:    $\mathbf{b}_2 \leftarrow$  Equation (3.131)

```

```

79:   |   |    $c \leftarrow$  Equation (3.134)
80:   |   |    $\Omega_f \leftarrow$  Equation (3.128)
81:   |   |    $\Omega_\tau \leftarrow$  Equation (3.129)
82:   |   |    $u_{eq} \leftarrow \Omega_\tau + \Omega_f$ 
83:   |   |    $s \leftarrow$  Equation (3.22)
84:   |   |    $u_{dis} \leftarrow$  Equation (3.27)
85:   |   |    $\Omega^d \leftarrow$  Equation (3.25)
86:   |   |    $\omega^d \leftarrow$  Equation (3.121)
87:   |   |   attCtrl  $\leftarrow \omega^d$ 
88: end case
89: case Adaptive (Section 3.4.3.1)
90:   |   |    $x, r, B_m, A_m, B_p \leftarrow$  Equation (3.70)
91:   |   |    $x_m \leftarrow$  Equation (3.48)
92:   |   |    $\Delta u \leftarrow$  Equation (3.52)
93:   |   |    $e_\Lambda \leftarrow$  Equation (3.56)
94:   |   |    $e \leftarrow$  Equation (3.54)
95:   |   |    $e_u \leftarrow$  Equation (3.57)
96:   |   |    $P \leftarrow$  previously calculated Equation (3.61)
97:   |   |    $\hat{K}_x \leftarrow$  Equation (3.62)
98:   |   |    $\hat{K}_r \leftarrow$  Equation (3.63)
99:   |   |    $\hat{f} \leftarrow$  Equation (3.64)
100:  |   |    $\hat{\lambda} \leftarrow$  Equation (3.65)
101:  |   |    $\Gamma_B^d \leftarrow$  Equation (3.49)
102:  |   |   attCtrl  $\leftarrow \Gamma_B^d$ 
103: end case
104: case Adaptive with PIDD (Section 3.4.3.2)
105:   |   |    $x, r, B_m, A_m, B_p, C \leftarrow$  Equation (3.74)
106:   |   |    $z_m \leftarrow$  Equation (3.72)
107:   |   |    $x_m \leftarrow$  Equation (3.73)
108:   |   |    $\Delta u \leftarrow$  Equation (3.52)
109:   |   |    $e_\Lambda \leftarrow$  Equation (3.56)
110:   |   |    $e \leftarrow$  Equation (3.54)
111:   |   |    $e_u \leftarrow$  Equation (3.57)
112:   |   |    $P \leftarrow$  previously calculated Equation (3.61)
113:   |   |    $\hat{K}_x \leftarrow$  Equation (3.62)
114:   |   |    $\hat{K}_r \leftarrow$  Equation (3.63)
115:   |   |    $\hat{f} \leftarrow$  Equation (3.64)
116:   |   |    $\hat{\lambda} \leftarrow$  Equation (3.65)
117:   |   |    $\Gamma_B^d, \mathbf{T}_A^{d*} \leftarrow$  Equation (3.49)

```

```

118: | attCtrl  $\leftarrow \boldsymbol{\Gamma}_B^d, \boldsymbol{T}_A^{d*}$ 
119: | end case
120: | case Adaptive Direct (Section 3.4.3.3)
121: |    $x, r, B_m, A_m, B_p, C \leftarrow$  Equation (3.75)
122: |    $z_m \leftarrow$  Equation (3.72)
123: |    $x_m \leftarrow$  Equation (3.73)
124: |    $\Delta u \leftarrow$  Equation (3.52)
125: |    $e_\Lambda \leftarrow$  Equation (3.56)
126: |    $e \leftarrow$  Equation (3.54)
127: |    $e_u \leftarrow$  Equation (3.57)
128: |    $P \leftarrow$  previously calculated Equation (3.61)
129: |    $\hat{K}_x \leftarrow$  Equation (3.62)
130: |    $\hat{K}_r \leftarrow$  Equation (3.63)
131: |    $\hat{f} \leftarrow$  Equation (3.64)
132: |    $\hat{\lambda} \leftarrow$  Equation (3.65)
133: |    $\boldsymbol{\Omega}^d \leftarrow$  Equation (3.49)
134: |    $\boldsymbol{\omega}^d \leftarrow$  Equation (3.121)
135: |   attCtrl  $\leftarrow \boldsymbol{\omega}^d$ 
136: | end case
137: | case R-LQR Passive (Section 3.4.4.1)
138: |    $a(t) \leftarrow$  Equation (2.25)
139: |    $A(\boldsymbol{\omega}_B, \omega_i) \leftarrow$  Equation (3.78)
140: |    $x_i, F_i, G_i \leftarrow$  Equation (3.82)
141: |    $u_i^* \leftarrow$  Steps 1 and 2 from Table 2
142: |    $\boldsymbol{\Gamma}_B^d \leftarrow$  Equation (3.82)
143: |   attCtrl  $\leftarrow \boldsymbol{\Gamma}_B^d$ 
144: | end case
145: | case R-LQR Active (Section 3.4.4.2)
146: |    $a_{fd}(t) \leftarrow$  Equations (3.35) and (4.9)
147: |    $A_{fd}(\boldsymbol{\omega}_B, \omega_i) \leftarrow$  Equation (3.83)
148: |    $x_i, F_i, G_i \leftarrow$  Equation (3.84)
149: |    $u_i^* \leftarrow$  Steps 1 and 2 from Table 2
150: |    $\boldsymbol{\Gamma}_B^d \leftarrow$  Equation (3.84)
151: |   attCtrl  $\leftarrow \boldsymbol{\Gamma}_B^d$ 
152: | end case
153: | case R-LQR Passive with rotor failures (Section 3.4.4.3)
154: |    $a(t) \leftarrow$  Equation (2.25)
155: |    $A(\boldsymbol{\omega}_B, \omega_i) \leftarrow$  Equation (3.78)
156: |    $x_i, F_i, G_i \leftarrow$  Equation (3.88)

```

```

157:            $u_i^* \leftarrow$  Steps 1 and 2 from Table 2
158:            $\Gamma_B^d \leftarrow$  Equation (3.88)
159:           attCtrl  $\leftarrow \Gamma_B^d$ 
160:       end case
161:   case R-LQR Active with rotor failures (Section 3.4.4.4)
162:        $\Lambda_{fda} \leftarrow$  Equations (3.45) and (4.10)
163:        $a_{fdd}(t) \leftarrow$  Equations (3.35) and (4.9)
164:        $A_{fdd}(\omega_B, \omega_i) \leftarrow$  Equation (3.83)
165:        $x_i, F_i, G_i \leftarrow$  Equation (3.89)
166:        $u_i^* \leftarrow$  Steps 1 and 2 from Table 2
167:        $\Gamma_B^d \leftarrow$  Equation (3.89)
168:       attCtrl  $\leftarrow \Gamma_B^d$ 
169:   end case
170:   case R-LQR Passive with rotor failures and PIDD (Section 3.4.4.5)
171:        $a(t) \leftarrow$  Equation (2.25)
172:        $A(\omega_B, \omega_i) \leftarrow$  Equation (3.78)
173:        $x_i, F_i, G_i \leftarrow$  Equation (3.90)
174:        $u_i^* \leftarrow$  Steps 1 and 2 from Table 2
175:        $\Gamma_B^d, \mathbf{T}_A^{d*} \leftarrow$  Equation (3.90)
176:       attCtrl  $\leftarrow \Gamma_B^d, \mathbf{T}_A^{d*}$ 
177:   end case
178:   case R-LQR Active with rotor failures and PIDD (Section 3.4.4.6)
179:        $\Lambda_{fda} \leftarrow$  Equations (3.45) and (4.10)
180:        $a_{fdd}(t) \leftarrow$  Equations (3.35) and (4.9)
181:        $A_{fdd}(\omega_B, \omega_i) \leftarrow$  Equation (3.83)
182:        $x_i, F_i, G_i \leftarrow$  Equation (3.91)
183:        $u_i^* \leftarrow$  Steps 1 and 2 from Table 2
184:        $\Gamma_B^d, \mathbf{T}_A^{d*} \leftarrow$  Equation (3.91)
185:       attCtrl  $\leftarrow \Gamma_B^d, \mathbf{T}_A^{d*}$ 
186:   end case
187:   case Mode-dependent R-LQR for DMJLSU (Section 3.4.5.1)
188:       for all  $\theta(k) \in \mathbb{S} \forall k$  do
189:            $a_{fdd}(t) \leftarrow$  Equations (3.35) and (4.9) for  $\Lambda_{\theta(k)}$ 
190:            $A_{fdd}(\omega_B, \omega_i) \leftarrow$  Equation (3.83) for  $\Lambda_{\theta(k)}$ 
191:            $F_{\theta(k)}, G_{\theta(k)} \leftarrow$  Equation (3.116)
192:       end for
193:        $x_k \leftarrow$  Equation (3.116)
194:        $K_{\theta(K),k} \leftarrow$  Step 1 from Table 4
195:        $\theta(k) = j \leftarrow \xi_{i,fdd}, \forall i$ 

```

```

196:   |   |    $u_k^* \leftarrow$  Step 2 from Table 4
197:   |   |    $\Gamma_B^d \leftarrow$  Equation (3.116)
198:   |   |   attCtrl  $\leftarrow \Gamma_B^d$ 
199:   |   |   end case
200:   |   |   end switch
201:   |   |   return attCtrl
202: end function

```

Algorithm 8 Control allocation methods from Section 3.5

```

1: function CONTROLALLOCATION(posCtrl, attCtrl, diagnosis)
2:   if  $\exists \Gamma_B^d$  and  $\mathbf{T}_A^{d*} \in$  attCtrl then
3:      $\Gamma_B^d, \mathbf{T}_A^{d*} \leftarrow$  attCtrl;
4:      $\mathbf{T}_B^d \leftarrow Q^{-1}\mathbf{T}_A^{d*};$ 
5:   else
6:      $\mathbf{T}_A^d \leftarrow$  posCtrl;
7:      $\mathbf{T}_B^d \leftarrow Q^{-1}\mathbf{T}_A^d;$ 
8:      $\Gamma_B^d \leftarrow$  attCtrl;
9:   end if
10:   $M_f \leftarrow$  Equation (3.8);
11:   $M_t \leftarrow$  Equation (3.43);
12:  ctrlAllocationType, motorDynamics  $\omega^{min}, \omega^{max}, W_m, W_a \leftarrow$  read values stored
    from previous user interface;
13:   $\xi_{i,fdd}, \xi_{P_i,fdd} \forall i = 1, \dots, N \leftarrow$  diagnosis;
14:  switch ctrlAllocationType do
15:    case PI Passive (Section 3.5.1)
16:       $H \leftarrow$  Equation (3.118);
17:       $\Omega^d \leftarrow$  Equation (3.119);
18:       $\omega^d \leftarrow$  Equation (3.121);
19:    end case
20:    case PI Active (Section 3.5.1.1)
21:       $\Lambda_{fdd} \leftarrow$  Equations (3.45) and (4.10);
22:       $H_{fdd} \leftarrow$  Equation (3.122);
23:       $\Omega^d \leftarrow$  Equation (3.119);
24:       $\omega^d \leftarrow$  Equation (3.121);
25:    end case
26:    case RPI Passive (Section 3.5.2)
27:       $H \leftarrow$  Equation (3.118);
28:       $\Omega^d \leftarrow$  Algorithm 1;
29:       $\omega^d \leftarrow$  Equation (3.121);

```

```

30:    end case
31:    case RPI Active (Section 3.5.2.1)
32:         $\Lambda_{fdd} \leftarrow$  Equations (3.45) and (4.10);
33:         $H_{fdd} \leftarrow$  Equation (3.122);
34:         $\omega_i^{min}, \omega_i^{max} \leftarrow$  Equations (3.125) and (3.126);
35:         $\Omega^d \leftarrow$  Algorithm 1;
36:         $\omega^d \leftarrow$  Equation (3.121);
37:    end case
38:    case NSCA Passive (Section 3.5.3)
39:         $\Omega_{1/2} \leftarrow$  Equation (??);
40:         $N_s \leftarrow \text{NULL}(M_t); \triangleright \text{NULL}(M_t)$  returns the null space base vectors of  $M_t$ 
41:         $b_2 \leftarrow$  Equation (3.131);
42:         $\Omega_\tau \leftarrow$  Equation (3.129);
43:         $c \leftarrow$  Equation (3.134);
44:         $\Omega_f \leftarrow$  Equation (3.128);
45:         $\Omega^d \leftarrow$  Equation (3.127);
46:         $\Omega' \leftarrow$  Equation (3.10);
47:         $\omega^d \leftarrow$  Equation (3.136);
48:    end case
49:    case NSCA Active (Section 3.5.3.1)
50:         $\Lambda_{fdd} \leftarrow$  Equations (3.45) and (4.10);
51:         $\omega^{min}, \omega^{max} \leftarrow$  Equations (3.125) and (3.126);
52:         $\Omega_{1/2} \leftarrow$  Equation (??);
53:         $N_s \leftarrow \text{NULL}(M_t \Lambda_{fdd});$ 
54:         $b_2 \leftarrow$  Equation (3.131);
55:         $\Omega_\tau \leftarrow$  Equation (3.129);
56:         $c \leftarrow$  Equation (3.134);
57:         $\Omega_f \leftarrow$  Equation (3.128);
58:         $\Omega^d \leftarrow$  Equation (3.127);
59:         $\Omega' \leftarrow$  Equation (3.10);
60:         $\omega^d \leftarrow$  Equation (3.136);
61:    end case
62:    case other or none
63:         $\omega^d \leftarrow \text{attCtrl};$ 
64:    end case
65: end switch
66: switch motorDynamics do
67:     case 'ON'
68:          $\text{actInput}_i \leftarrow \frac{R_i C_{Diop} \omega_i^d |\omega_i^d|}{\kappa_\tau} + \kappa_v \omega_i^d, \text{ for } i = 1, \dots, N;$ 

```

```

69:    |   end case
70:    |   case 'OFF'
71:    |       actInput ←  $\omega^d$ ;
72:    |   end case
73:   end switch
74:   return actInput;
75: end function

```

Algorithm 9 Method "run" of class @multicopter to simulate multirotor dynamics

```

1: function RUN@MULTICOPTER(actInput, inputTime, time)
2:    $\delta t_{sim}$  ← Simulation step period;
3:   prevActInput ← last actuator input from previous iteration or initial value;
4:   prevTime ← last simulation time from previous iteration or initial value;
5:   simTime ← prevTime: $\delta t_{sim}$ :time;
6:   simInput ← resample [prevActInput, actinput] over  $\delta t_{sim}$  such that
7:   LENGTH(simTime) == LENGTH(simInput) and inputTime defines the transition
8:   time from prevActInput to actInput;
9:   solver ← chosen solver for simulation;
10:   $y_0$  ← last states from previous iteration or initial states;
11:  switch solver do
12:    |   case ode45
13:      |       if is first run then
14:          |           |   solver ← ODE45(@(t,y) MODEL(t,y,simTime,simInput),
15:          |           |   [simTime(1) simTime(end)],  $y_0$ );    ▷ ODE45 solves the model from
16:          |           |   time simTime(1) to simTime(end) on initial states  $y_0$ 
17:          |       else
18:          |           |   solver ← ODEXTEND(solver, @(t,y) MODEL(t,y,simTime,simInput),
19:          |           |   simTime(end));            ▷ ODEXTEND does the same as ODE45 but
20:          |           |   considering values saved on previous iterations
21:          |       end if
22:          |       outputStates ← DEVAL(solver,simTime);    ▷ DEVAL evaluates the ODE
23:          |       solution over the points simTime
24:    |   end case
25:    |   case ode15s
26:      |       if is first run then
27:          |           |   solver ← ODE15S(@(t,y) MODEL(t,y,simTime,simInput),
28:          |           |   [simTime(1) simTime(end)],  $y_0$ );    ▷ ODE15S solves the model from
29:          |           |   time simTime(1) to simTime(end) on initial states  $y_0$ 
30:      |       else

```

```

31:     solver ← ODEXTEND(solver, @(t,y) MODEL(t,y,simTime,simInput),
32:                         simTime(end));           ▷ ODEXTEND does the same as ODE15S but
33:                         considering values saved on previous iterations
34:     end if
35:     outputStates ← DEVAL(solver,simTime);
36:   end case
37:   case euler
38:     outputStates(:,1) ←  $y_0$ ;
39:     for it=1:LENGTH(simTime)-1 do
40:       dydt ← MODEL(simTime(it),outputStates(:,it),simTime,simInput);
41:       outputStates(:,it+1) ← outputStates(:,it) +  $\delta t_{sim}$  dydt;
42:       q ← outputStates;
43:       q ← q/|q|;
44:       outputStates ← q;
45:     end for
46:   end case
47:   case midpoint
48:     outputStates(:,1) ←  $y_0$ ;
49:     for it=1:LENGTH(simTime)-1 do
50:       aux ← MODEL(simTime(it), outputStates(:,it), simTime, simInput);
51:       dydt ← MODEL(simTime(it)+0.5 $\delta t_{sim}$ , outputStates(:,it)+0.5 $\delta t_{sim}$ 
52:                   aux, simTime, simInput);
53:       outputStates(:,it+1) ← outputStates(:,it) +  $\delta t_{sim}$  dydt;
54:       q ← outputStates;
55:       q ← q/|q|;
56:       outputStates ← q;
57:     end for
58:   end case
59: end switch
60: currState ← outputStates(:,end);
61: return currState;
62: end function

```

Algorithm 10 Implementation of model from Chapter 2

```

1: function MODEL(evaluationTime, initialStates, simulationTime, simulationInput)
2:   motorDynamics,  $v_{LLi_{max}}$  ← read values stored from previous user interface;
3:   t ← evaluationTime;
4:   auxIndex ← FIND( $t \leq$  simTime);    ▷ FIND( $t \leq$  simTime) returns the indexes of
5:   simTime that satisfy  $t \leq$  simTime;

```

```

6:   input ← simInput(:,auxIndex);
7:    $\xi_i, \xi_{Pi}$  for  $i = 1, \dots, N$  ← Failure simulation from simulator command;
8:    $I_{cg} \leftarrow$  Equation (4.15);
9:    $m \leftarrow$  Equation (4.16);
10:   $r_i, i = 1, \dots, N \leftarrow$  Equation (4.17);
11:  switch motorDynamics do
12:    case 'ON'
13:      for  $doi = 1, \dots, N$ 
14:         $v_{LLi} \leftarrow$  input;
15:         $v_{LLi} \leftarrow \text{SIGN}(v_{LLi}) \text{ MIN}(|v_{LLi}|, v_{LLi_{max}})$ ;
16:         $\dot{\omega}_i \leftarrow$  Equation (4.7);
17:      end for
18:    end case
19:    case 'OFF'
20:      for  $doi = 1, \dots, N$ 
21:         $\omega_i^d \leftarrow$  input;
22:         $\omega_i^{d'} \leftarrow$  Equation (3.10);
23:         $\omega_i \leftarrow \omega_i^{d'}$ ;
24:      end for
25:    end case
26:  end switch
27:   $q, \mathbf{v}_A, \boldsymbol{\omega}_B \leftarrow$  initialStates;
28:   $S(q) \leftarrow$  Equation (2.28);
29:   $Q \leftarrow$  Equation (2.6);
30:   $\dot{\boldsymbol{\omega}}_B \leftarrow$  Equation (2.24);
31:   $\mathbf{f}_{dist}(t) \leftarrow$  Equation (4.19);
32:   $\dot{\mathbf{v}}_A \leftarrow$  Equation (4.18);
33:   $\dot{q} \leftarrow$  Equation (2.27);
34:   $\dot{\mathbf{P}}_A \leftarrow$  Equation (2.11);
35:  switch motorDynamics do
36:    case 'ON'
37:      stateDerivatives  $\leftarrow \dot{\mathbf{P}}_A, \dot{q}, \dot{\mathbf{v}}_A, \dot{\boldsymbol{\omega}}_B, \dot{\omega}_i$  for  $i = 1, \dots, N$ ;
38:    end case
39:    case 'OFF'
40:      stateDerivatives  $\leftarrow \dot{\mathbf{P}}_A, \dot{q}, \dot{\mathbf{v}}_A, \dot{\boldsymbol{\omega}}_B$ ;
41:    end case
42:  end switch
43:  return stateDerivatives;
44: end function

```

APPENDIX E – BEST INDIVIDUALS AFTER CONTROLLER OPTIMIZATIONS

This appendix comprises all the controller cases parameters after optimization using the genetic algorithm proposed in Section 4.4. Variables notation follow the one used in the controllers definition in Chapter 3.

E.1 Case 1.1.1

$$K_P = \begin{pmatrix} 115.6 & 139.9 & 71.55 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 10.0 & 10.0 & 134.4 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 20.0 & 36.34 & 20.0 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

$$K_{\tau P} = \begin{pmatrix} 996.5 & 984.6 & 199.0 \end{pmatrix}$$

$$K_{\tau I} = \begin{pmatrix} 972.5 & 984.4 & 23.1 \end{pmatrix}$$

$$K_{\tau D} = \begin{pmatrix} 44.72 & 37.22 & 10.0 \end{pmatrix}$$

$$\alpha = \text{diag} \begin{pmatrix} 0.9175 & 0 & 0.6534 \end{pmatrix} \text{ from Equation (4.1)}$$

E.2 Case 1.1.2.1

$$K_P = \begin{pmatrix} 154.5 & 161.3 & 86.1 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 0.5154 & 0.9925 & 174.1 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 20.0 & 20.0 & 19.81 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

$$c = \begin{pmatrix} 29.49 & 0 & 0 \\ 0 & 54.15 & 0 \\ 0 & 0 & 18.4 \end{pmatrix}$$

$$\lambda = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 77.58 \end{pmatrix}$$

$$\alpha = \begin{pmatrix} 5.92 & 0 & 0 \\ 0 & 8.586 & 0 \\ 0 & 0 & 2.0 \end{pmatrix}$$

$$\alpha = \text{diag} \begin{pmatrix} 0.2619 & 0.4232 & 0.948 \end{pmatrix} \text{ from Equation (4.1)}$$

E.3 Case 1.1.2.2

$$K_P = \begin{pmatrix} 118.5 & 128.7 & 78.91 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 0.3298 & 0.3447 & 145.3 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 20.0 & 21.11 & 14.44 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0.06928 & 0 \end{pmatrix}$$

$$c = \begin{pmatrix} 35.12 & 0 & 0 & 0 & 0 & 0 \\ 0 & 53.95 & 0 & 0 & 0 & 0 \\ 0 & 0 & 17.86 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 49.69 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9633 \end{pmatrix}$$

$$\lambda = \begin{pmatrix} 0.1213 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2548 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 80.41 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10.03 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.926 \end{pmatrix}$$

$$\alpha = \begin{pmatrix} 4.641 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.83 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.115 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.024 & 0 & 0 \\ 0 & 0 & 0 & 0 & 60.95 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.36 \end{pmatrix}$$

$\alpha = \text{diag} \begin{pmatrix} 0.353 & 0.4353 & 0.04448 \end{pmatrix}$ from Equation (4.1)

E.4 Case 1.1.2.3

$$K_P = \begin{pmatrix} 95.58 & 83.9 & 33.11 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 5.431 & 1.029 & 57.98 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 20.0 & 20.94 & 20.0 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

$$\begin{aligned}
c &= \begin{pmatrix} 32.44 & 0 & 0 & 0 & 0 & 0 \\ 0 & 28.65 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.716 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 64.15 \end{pmatrix} \\
\lambda &= \begin{pmatrix} 4000.0 & -400.0 & 4.0 & 4000.0 & 38977.0 & 8552.0 \\ -2298.0 & 4.0 & 4000.0 & -4000.0 & 48600.0 & 12000.0 \\ 400.0 & -4.0 & -10788.0 & -400.0 & -4.0 & -1134.0 \\ -7017.0 & 4000.0 & -4000.0 & -4.0 & 4980.0 & -8496.0 \\ -4.0 & -35911.0 & -37933.0 & -41100.0 & -47166.0 & 400.0 \\ 5773.0 & 21100.0 & -39988.0 & 11444.0 & 48999.0 & -17866.0 \\ 996.9 & 37677.0 & 43288.0 & 43300.0 & -400.0 & 4.0 \\ 630.0 & 4.0 & 46222.0 & 4000.0 & 4.0 & 4000.0 \end{pmatrix} \\
\alpha &= \begin{pmatrix} 9.242 & -39300.0 & 20600.0 & 42100.0 & 48722.0 & 55.0 \\ -5500.0 & 13366.0 & 5500.0 & -34488.0 & 45422.0 & 5500.0 \\ 550.0 & -55.0 & -1571.0 & -9482.0 & -55.0 & -5500.0 \\ -49800.0 & 49999.0 & -5500.0 & -5.5 & 2553.0 & -5500.0 \\ -55.0 & -5500.0 & -1975.0 & -55.0 & -5500.0 & 18466.0 \\ 27133.0 & 14633.0 & -26177.0 & 5500.0 & 27488.0 & -5.5 \\ 9903.0 & 14877.0 & 6342.0 & 5500.0 & -46777.0 & 1439.0 \\ 14633.0 & 5.5 & 271.5 & 5500.0 & 5.5 & 5500.0 \end{pmatrix} \\
\alpha &= \text{diag} \left(\begin{array}{ccc} 0 & 0.2528 & 0.5 \end{array} \right) \text{ from Equation (4.1)}
\end{aligned}$$

E.5 Case 1.1.3.1

$$K_P = \begin{pmatrix} 124.4 & 135.0 & 40.0 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 10.0 & 4.444 & 74.89 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 20.0 & 20.0 & 13.97 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

$$P = \begin{pmatrix} 4.712 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9.914 \cdot 10^6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.064 \cdot 10^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9.863 \cdot 10^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8.612 \cdot 10^5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.251 \cdot 10^6 \end{pmatrix}$$

$$Q = \begin{pmatrix} 1.062 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6.792 \cdot 10^6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8.634 \cdot 10^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.859 \cdot 10^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9.765 \cdot 10^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8.689 \cdot 10^6 \end{pmatrix}$$

$$R = \begin{pmatrix} 70.0 & 0 & 0 \\ 0 & 70.0 & 0 \\ 0 & 0 & 7.0 \end{pmatrix}$$

$$E_f = \begin{pmatrix} 81.55 & 6.495 & 143.4 & 957.0 & 671.5 & 0 \end{pmatrix}$$

$$E_g = \begin{pmatrix} 0.02 & 0.02 & 103.4 \end{pmatrix}$$

$$H = \begin{pmatrix} 371.9 \\ 7.06 \\ 1.0 \\ 1.0 \\ 941.4 \\ 335.8 \end{pmatrix}$$

$$\mu = 3.188 \cdot 10^{29}$$

$$\alpha = 1.5$$

$$\alpha = \text{diag} \begin{pmatrix} 0 & 0 & 0.06539 \end{pmatrix} \text{ from Equation (4.1)}$$

E.6 Case 1.1.3.2

$$K_P = \begin{pmatrix} 135.0 & 141.1 & 40.0 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 1.828 & 10.0 & 113.8 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 20.0 & 20.0 & 20.0 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0 & 1.025 \end{pmatrix}$$

$$P = \begin{pmatrix} 5.0 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.0 \cdot 10^6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.0 \cdot 10^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9.676 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.0 \cdot 10^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.0 \cdot 10^6 \end{pmatrix}$$

$$Q = \begin{pmatrix} 5.0 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.0 \cdot 10^6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.711 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.604 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8.295 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9.081 \cdot 10^7 \end{pmatrix}$$

$$R = \begin{pmatrix} 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 351.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} \end{pmatrix}$$

$$E_f = \begin{pmatrix} 79144.0 & 85266.0 & 77077.0 & 12544.0 & 77577.0 & 0 \end{pmatrix}$$

$$E_g = \begin{pmatrix} 1000.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 \end{pmatrix}$$

$$H = \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{pmatrix}$$

$$\mu = 1.0 \cdot 10^{20}$$

$$\alpha = 26.54$$

$$\alpha = \text{diag} \begin{pmatrix} 0 & 0.1194 & 0.8995 \end{pmatrix} \text{ from Equation (4.1)}$$

E.7 Case 1.1.3.3

$$K_P = \begin{pmatrix} 107.9 & 80.9 & 40.0 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 0.1551 & 1.23 & 102.5 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 20.94 & 20.0 & 20.0 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

$$P = \begin{pmatrix} 9.682 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.927 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.0 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.508 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7.192 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.56 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7.14 \cdot 10^7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.42 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5.0 \cdot 10^6 \end{pmatrix}$$

$$Q = \begin{pmatrix} 9.215 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.034 \cdot 10^7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.225 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.0 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.0 \cdot 10^6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.798 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9.226 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5.273 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.149 \cdot 10^8 \end{pmatrix}$$

$$R = \begin{pmatrix} 580.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 \end{pmatrix}$$

$$E_f = \begin{pmatrix} 40111.0 & 10.0 & 10.0 & 61366.0 & 10.0 & 2794.0 & 31600.0 & 62988.0 & 0 \end{pmatrix}$$

$$E_g = \begin{pmatrix} 78322.0 & 40.0 & 40.0 & 40.0 & 2601.0 & 40.0 & 40.0 & 40.0 \end{pmatrix}$$

$$H = \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{pmatrix}$$

$$\mu = 1.0 \cdot 10^{22}$$

$$\alpha = 1.5$$

$\alpha = \text{diag} \begin{pmatrix} 0.1301 & 0.0514 & 0.9006 \end{pmatrix}$ from Equation (4.1)

E.8 Case 1.1.4

$$K_P = \begin{pmatrix} 69.47 & 74.28 & 9.886 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 1.892 & 4.53 & 206.9 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 29.03 & 14.15 & 44.89 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0.1051 & 0.3441 \end{pmatrix}$$

$$E_{f1} = \begin{pmatrix} 99233.0 & 6595.0 & 30666.0 & 64200.0 & 75500.0 & 10.0 \end{pmatrix}$$

$$E_{f2} = \begin{pmatrix} 60722.0 & 10.0 & 92799.0 & 10.0 & 99522.0 & 10.0 \end{pmatrix}$$

$$E_{f3} = \begin{pmatrix} 31844.0 & 10.0 & 83800.0 & 54799.0 & 10.0 & 10.0 \end{pmatrix}$$

$$E_{f4} = \begin{pmatrix} 42900.0 & 10.0 & 10.0 & 94399.0 & 10.0 & 16400.0 \end{pmatrix}$$

$$E_{f5} = \begin{pmatrix} 10.0 & 21700.0 & 35299.0 & 64211.0 & 58733.0 & 63388.0 \end{pmatrix}$$

$$E_{g1} = \begin{pmatrix} 48699.0 & 35688.0 & 93677.0 & 96188.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 \end{pmatrix}$$

$$E_{g2} = \begin{pmatrix} 1000.0 & 1072.0 & 1000.0 & 79399.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 \end{pmatrix}$$

$$E_{o3} = \begin{pmatrix} 13033.0 & 6197.0 & 83188.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 \end{pmatrix}$$

$$E_{q4} = \begin{pmatrix} 1000.0 & 1000.0 & 2481.0 & 77200.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 \end{pmatrix}$$

$$E_{55} \equiv \begin{pmatrix} 51599.0 & 46122.0 & 66500.0 & 1000.0 & 34277.0 & 1000.0 & 12233.0 & 100.0 \end{pmatrix}$$

$$\begin{aligned}
E_{r2} &= \begin{pmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.04835 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.7775 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} \end{pmatrix} \\
E_{r3} &= \begin{pmatrix} 0.4154 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9594 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 \end{pmatrix} \\
E_{r4} &= \begin{pmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2009 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.223 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4016 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 \end{pmatrix} \\
E_{r5} &= \begin{pmatrix} 0.3002 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.03051 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8049 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.6086 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 \end{pmatrix} \\
E_{q1} &= \begin{pmatrix} 94633.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 97799.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1747.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 93100.0 \end{pmatrix}
\end{aligned}$$

$$E_{q2} = \begin{pmatrix} 10344.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 37222.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 47499.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 62800.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} \end{pmatrix}$$

$$E_{q3} = \begin{pmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20344.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1055.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 96511.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8237.0 \end{pmatrix}$$

$$E_{q4} = \begin{pmatrix} 47122.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 74544.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 32322.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 46866.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} \end{pmatrix}$$

$$E_{q5} = \begin{pmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 54299.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 6801.0 \end{pmatrix}$$

$$\lambda = 1.0$$

$$modes = \begin{pmatrix} 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 0 & 0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 0 & 0 & 0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \end{pmatrix}$$

$$numberOfModes = 5.0$$

$$p_{ij} = \begin{pmatrix} 6.488 & 0 & 0 & 0 & 0 \\ 0 & 6.488 & 0 & 0 & 0 \\ 0 & 0 & 6.488 & 0 & 0 \\ 0 & 0 & 0 & 6.488 & 0 \\ 0 & 0 & 0 & 0 & 6.488 \end{pmatrix}$$

$$e_{ij} = \begin{pmatrix} 2.0 & 2.0 & 2.0 & 2.0 & 2.0 \\ 2.0 & 2.0 & 2.0 & 2.0 & 2.0 \\ 2.0 & 2.0 & 2.0 & 2.0 & 2.0 \\ 2.0 & 2.0 & 2.0 & 2.0 & 2.0 \\ 2.0 & 2.0 & 2.0 & 2.0 & 2.0 \end{pmatrix}$$

$$\alpha = \text{diag} \left(\begin{array}{ccc} 0 & 0 & 0.2465 \end{array} \right) \text{ from Equation (4.1)}$$

E.9 Case 1.1.5.1

$$K_P = \left(\begin{array}{ccc} 64.24 & 68.42 & 261.0 \end{array} \right)$$

$$K_I = \left(\begin{array}{ccc} 0 & 0.3489 & 646.0 \end{array} \right)$$

$$K_D = \left(\begin{array}{ccc} 5.0 & 17.29 & 25.78 \end{array} \right)$$

$$K_{DD} = \left(\begin{array}{ccc} 0 & 0 & 0 \end{array} \right)$$

$$A_m = \begin{pmatrix} -0.2 & 0 & 0 \\ 0 & -0.2 & 0 \\ 0 & 0 & -0.09469 \end{pmatrix}$$

$$Q = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{pmatrix}$$

$$\gamma_1 = \begin{pmatrix} 0.8584 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.001 \end{pmatrix}$$

$$\gamma_2 = \begin{pmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.001 \end{pmatrix}$$

$$\gamma_3 = \begin{pmatrix} 13.69 & 0 & 0 \\ 0 & 12.43 & 0 \\ 0 & 0 & 4.0 \end{pmatrix}$$

$$\gamma_4 = \begin{pmatrix} 0.0001 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & 12.2 \end{pmatrix}$$

$$\alpha = \text{diag} \left(\begin{array}{ccc} 0.005234 & 0 & 0.4284 \end{array} \right) \text{ from Equation (4.1)}$$

E.10 Case 1.1.5.2

$$K_P = \left(\begin{array}{ccc} 64.24 & 81.72 & 40.0 \end{array} \right)$$

$$K_I = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 13.64 & 5.0 & 5.0 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0 & 0.04845 \end{pmatrix}$$

$$A_m = \begin{pmatrix} -0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -12.92 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.01 \end{pmatrix}$$

$$Q = \begin{pmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.541 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.442 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0005 \end{pmatrix}$$

$$\gamma_1 = \begin{pmatrix} 0.0001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.122 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0001 \end{pmatrix}$$

$$\gamma_2 = \begin{pmatrix} 0.0001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0001 \end{pmatrix}$$

$$\gamma_3 = \begin{pmatrix} 4.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7.477 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.981 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.681 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.775 \end{pmatrix}$$

$$\gamma_4 = \begin{pmatrix} 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 16.85 \end{pmatrix}$$

$\alpha = \text{diag} \begin{pmatrix} 0 & 0 & 0.2171 \end{pmatrix}$ from Equation (4.1)

E.11 Case 1.1.5.3

$$K_P = \begin{pmatrix} 30.0 & 30.0 & 40.0 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 0 & 0 & 56.25 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 5.0 & 5.0 & 5.0 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

$$A_m = \begin{pmatrix} -86.99 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.07152 & 0 & 0 \\ 0 & 0 & 0 & 0 & -84.86 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.001 \end{pmatrix}$$

$$Q = \begin{pmatrix} 4.851 & 0 & 0 & 0 & 0 & 0 \\ 0 & 35.52 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0005 & 0 & 0 \\ 0 & 0 & 0 & 0 & 49.9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.814 \end{pmatrix}$$

$$\gamma_1 = \begin{pmatrix} 1347.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 987.9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4181.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2493.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 531.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6132.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10000.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6491.0 \end{pmatrix}$$

$$\gamma_2 = \begin{pmatrix} 3867.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 200.9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4616.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1272.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4759.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3336.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10000.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1453.0 \end{pmatrix}$$

$$\gamma_3 = \begin{pmatrix} 10000.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3240.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3823.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7376.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 407.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9497.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8046.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4697.0 \end{pmatrix}$$

$$\gamma_4 = \begin{pmatrix} 0.005 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.005 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.005 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.005 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.005 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.005 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.005 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.005 \end{pmatrix}$$

$$B_0 = \begin{pmatrix} -12000.0 & 1625.0 & 345.2 & -896.6 & 6954.0 & 7500.0 \\ -24077.0 & 9609.0 & -22766.0 & -0.05 & 633.2 & 15000.0 \\ 20000.0 & 21388.0 & -6797.0 & 22766.0 & 19000.0 & -17277.0 \\ 7500.0 & 10266.0 & 0.05 & 7500.0 & -15000.0 & 16600.0 \\ -8356.0 & 21599.0 & 2438.0 & -4354.0 & 0.05 & 10577.0 \\ -5880.0 & 0.05 & -7500.0 & 11311.0 & 9032.0 & -13588.0 \\ 24811.0 & 54.91 & -1588.0 & 24644.0 & 23822.0 & -20722.0 \\ -21844.0 & -4.902 & 274.7 & -21511.0 & -12144.0 & -25000.0 \end{pmatrix}$$

$$\alpha = \text{diag} \left(\begin{array}{ccc} 0.9128 & 0.8962 & 0.5 \end{array} \right) \text{ from Equation (4.1)}$$

E.12 Case 1.2.1

$$K_P = \text{diag} \left(\begin{array}{ccc} 90.0 & 90.23 & 90.0 \end{array} \right)$$

$$K_I = \text{diag} \left(\begin{array}{ccc} 9.762 & 4.265 & 219.3 \end{array} \right)$$

$$\begin{aligned}
K_D &= \text{diag} \left(\begin{array}{ccc} 41.77 & 37.41 & 40.0 \end{array} \right) \\
K_{DD} &= \text{diag} \left(\begin{array}{ccc} 4.108 & 3.247 & 6.795 \end{array} \right) \\
K_{\tau P} &= \text{diag} \left(\begin{array}{ccc} 130.1 & 131.5 & 45.14 \end{array} \right) \\
K_{\tau I} &= \text{diag} \left(\begin{array}{ccc} 280.1 & 475.1 & 1.489 \end{array} \right) \\
K_{\tau D} &= \text{diag} \left(\begin{array}{ccc} 14.45 & 18.78 & 2.0 \end{array} \right) \\
\alpha &= \text{diag} \left(\begin{array}{ccc} 0.853 & 0.3896 & 0.5 \end{array} \right) \text{ from Equation (4.1)}
\end{aligned}$$

E.13 Case 1.2.2.1

$$\begin{aligned}
K_P &= \text{diag} \left(\begin{array}{ccc} 191.8 & 148.3 & 782.5 \end{array} \right) \\
K_I &= \text{diag} \left(\begin{array}{ccc} 21.99 & 2.731 & 65.28 \end{array} \right) \\
K_D &= \text{diag} \left(\begin{array}{ccc} 45.54 & 37.58 & 70.0 \end{array} \right) \\
K_{DD} &= \text{diag} \left(\begin{array}{ccc} 11.9 & 16.15 & 27.6 \end{array} \right) \\
c &= \begin{pmatrix} 21.43 & 0 & 0 \\ 0 & 27.02 & 0 \\ 0 & 0 & 3.0 \end{pmatrix} \\
\lambda &= \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 19.35 \end{pmatrix} \\
\alpha &= \begin{pmatrix} 2.0 & 0 & 0 \\ 0 & 4.153 & 0 \\ 0 & 0 & 35.02 \end{pmatrix} \\
\alpha &= \text{diag} \left(\begin{array}{ccc} 0.9197 & 0.938 & 0.9482 \end{array} \right) \text{ from Equation (4.1)}
\end{aligned}$$

E.14 Case 1.2.2.2

$$\begin{aligned}
K_P &= \text{diag} \left(\begin{array}{ccc} 51.94 & 125.4 & 467.7 \end{array} \right) \\
K_I &= \text{diag} \left(\begin{array}{ccc} 121.5 & 2.371 & 920.4 \end{array} \right) \\
K_D &= \text{diag} \left(\begin{array}{ccc} 45.54 & 22.92 & 70.0 \end{array} \right) \\
K_{DD} &= \text{diag} \left(\begin{array}{ccc} 1.853 & 2.866 & 17.68 \end{array} \right) \\
c &= \text{diag} \left(\begin{array}{cccccc} 0.2376 & 0.798 & 3.0 & 83.55 & 85.74 & 6.0 \end{array} \right) \\
\lambda &= \text{diag} \left(\begin{array}{cccccc} 0.1 & 0.1 & 0.3059 & 3.523 & 51.24 & 0.2 \end{array} \right) \\
\alpha &= \text{diag} \left(\begin{array}{cccccc} 2.0 & 4.438 & 2.86 & 39.89 & 97.71 & 4.0 \end{array} \right)
\end{aligned}$$

$$\alpha = \text{diag} \begin{pmatrix} 0 & 0 & 0.821 \end{pmatrix} \text{ from Equation (4.1)}$$

E.15 Case 1.2.2.3

$$K_P = \text{diag} \begin{pmatrix} 175.0 & 300.0 & 100.0 \end{pmatrix}$$

$$K_I = \text{diag} \begin{pmatrix} 30.51 & 10.0 & 56.31 \end{pmatrix}$$

$$K_D = \text{diag} \begin{pmatrix} 45.54 & 67.17 & 70.0 \end{pmatrix}$$

$$K_{DD} = \text{diag} \begin{pmatrix} 14.96 & 19.89 & 14.54 \end{pmatrix}$$

$$c = \text{diag} \begin{pmatrix} 1.5 & 1.5 & 6.0 & 3.0 & 3.0 & 3.0 \end{pmatrix}$$

$$\lambda = \begin{pmatrix} 1000.0 & -100.0 & 1.0 & 1000.0 & 100.0 & 1.0 \\ -1000.0 & 1.0 & 37144.0 & -1000.0 & 1.0 & 1000.0 \\ 100.0 & -1.0 & -1000.0 & -6375.0 & -1.0 & -1000.0 \\ -1.0 & 1000.0 & -1000.0 & -1.0 & 26788.0 & -1000.0 \\ -1.0 & -1000.0 & -100.0 & -1.0 & -1000.0 & 2232.0 \\ 1000.0 & 1000.0 & -1.0 & 1000.0 & 1000.0 & -21777.0 \\ 1000.0 & 100.0 & 1.0 & 1000.0 & -100.0 & 1369.0 \\ 1000.0 & 1.0 & 26800.0 & 1000.0 & 5426.0 & 1000.0 \end{pmatrix}$$

$$\alpha = \begin{pmatrix} 10688.0 & -45322.0 & 55.0 & 5500.0 & 550.0 & 55.0 \\ -5500.0 & 5.5 & 5500.0 & -49977.0 & 11566.0 & 5500.0 \\ 550.0 & -55.0 & -32744.0 & -22333.0 & -18266.0 & -34822.0 \\ -33188.0 & 5500.0 & -39411.0 & -38000.0 & 5500.0 & -5500.0 \\ -55.0 & -5500.0 & -550.0 & -55.0 & -5500.0 & 22611.0 \\ 25444.0 & 5500.0 & -5.5 & 5500.0 & 5500.0 & -5.5 \\ 5500.0 & 550.0 & 55.0 & 5500.0 & -550.0 & 11388.0 \\ 3640.0 & 5.5 & 5500.0 & 5500.0 & 5.5 & 5500.0 \end{pmatrix}$$

$$\alpha = \text{diag} \begin{pmatrix} 0 & 0 & 0.9505 \end{pmatrix} \text{ from Equation (4.1)}$$

E.16 Case 1.2.3.1

$$K_P = \text{diag} \begin{pmatrix} 1.388 & 1.036 & 506.6 \end{pmatrix}$$

$$K_I = \text{diag} \begin{pmatrix} 663.1 & 69.81 & 992.1 \end{pmatrix}$$

$$K_D = \text{diag} \begin{pmatrix} 55.85 & 6.395 & 95.4 \end{pmatrix}$$

$$K_{DD} = \text{diag} \begin{pmatrix} 6.315 & 13.03 & 17.94 \end{pmatrix}$$

$$P = \text{diag} \begin{pmatrix} 6.603 \cdot 10^6 & 9.263 \cdot 10^6 & 2.612 \cdot 10^6 & 4.983 \cdot 10^6 & 1.532 \cdot 10^6 & 9.053 \cdot 10^6 \end{pmatrix}$$

$$Q = \text{diag} \begin{pmatrix} 8.887 \cdot 10^6 & 2.492 \cdot 10^5 & 6.379 \cdot 10^6 & 7.962 \cdot 10^6 & 6.259 \cdot 10^6 & 2.421 \cdot 10^6 \end{pmatrix}$$

$$R = \text{diag} \left(\begin{array}{ccc} 2.207 \cdot 10^6 & 9.851 \cdot 10^6 & 7.821 \cdot 10^6 \end{array} \right)$$

$$E_f = \left(\begin{array}{cccccc} 483.2 & 168.5 & 532.9 & 860.8 & 130.7 & 71.74 \end{array} \right)$$

$$E_g = \left(\begin{array}{ccc} 6.831 & 21.11 & 123.9 \end{array} \right)$$

$$H = \begin{pmatrix} 41.94 \\ 603.7 \\ 145.5 \\ 27.06 \\ 138.6 \\ 660.8 \end{pmatrix}$$

$$\mu = 2.009 \cdot 10^{28}$$

$$\alpha = 75.99$$

$$\alpha = \text{diag} \left(\begin{array}{ccc} 0.9958 & 0.02262 & 0.8353 \end{array} \right) \text{ from Equation (4.1).}$$

E.17 Case 1.2.3.2

$$K_P = \text{diag} \left(\begin{array}{ccc} 136.0 & 153.1 & 60.0 \end{array} \right)$$

$$K_I = \text{diag} \left(\begin{array}{ccc} 2.0 & 2.0 & 67.62 \end{array} \right)$$

$$K_D = \text{diag} \left(\begin{array}{ccc} 25.0 & 42.66 & 30.0 \end{array} \right)$$

$$K_{DD} = \text{diag} \left(\begin{array}{ccc} 3.0 & 3.0 & 3.0 \end{array} \right)$$

$$P = \text{diag} \left(\begin{array}{cccccc} 1.364 \cdot 10^8 & 1.653 \cdot 10^8 & 3.38 \cdot 10^8 & 8.117 \cdot 10^8 & 4.707 \cdot 10^8 & 5.0 \cdot 10^6 \end{array} \right)$$

$$Q = \text{diag} \left(\begin{array}{cccccc} 5.0 \cdot 10^6 & 5.0 \cdot 10^6 & 5.0 \cdot 10^6 & 5.174 \cdot 10^7 & 1.324 \cdot 10^7 & 5.0 \cdot 10^6 \end{array} \right)$$

$$R = \text{diag} \left(\begin{array}{cccccccc} 582.1 & 1.0 \cdot 10^{-5} & 1.0 \cdot 10^{-5} & 1.0 \cdot 10^{-5} & 169.7 & 1.0 \cdot 10^{-5} & 1.0 \cdot 10^{-5} & 42.8 \end{array} \right)$$

$$E_f = \left(\begin{array}{ccccccc} 64244.0 & 35822.0 & 10.0 & 56099.0 & 0 & 0 \end{array} \right)$$

$$E_g = \left(\begin{array}{cccccccc} 15588.0 & 36288.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 \end{array} \right)$$

$$H = \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{pmatrix}$$

$$\mu = 1.0 \cdot 10^{20}$$

$$\alpha = 1.5$$

$$\alpha = \text{diag} \left(\begin{array}{ccc} 0.894 & 0.8669 & 0.7133 \end{array} \right) \text{ from Equation (4.1)}$$

E.18 Case 1.2.3.3

$$K_P = \text{diag} \begin{pmatrix} 70.0 & 70.0 & 100.0 \end{pmatrix}$$

$$K_I = \text{diag} \begin{pmatrix} 5.0 & 5.0 & 40.0 \end{pmatrix}$$

$$K_D = \text{diag} \begin{pmatrix} 40.0 & 40.0 & 30.0 \end{pmatrix}$$

$$K_{DD} = \text{diag} \begin{pmatrix} 2.0 & 2.0 & 1.0 \end{pmatrix}$$

$$P = \text{diag} \begin{pmatrix} 5.0 \cdot 10^6 & 8.866 \cdot 10^8 \end{pmatrix}$$

$$R = \text{diag} \begin{pmatrix} 1.0 \cdot 10^{-5} & 1.0 \cdot 10^{-5} & 1.0 \cdot 10^{-5} & 1.0 \cdot 10^{-5} & 245.3 & 1.0 \cdot 10^{-5} & 1.0 \cdot 10^{-5} & 1.0 \cdot 10^{-5} \end{pmatrix}$$

$$E_f = \begin{pmatrix} 10.0 & 76111.0 & 10.0 & 10.0 & 10.0 & 10.0 & 0 & 0 & 0 \end{pmatrix}$$

$$E_g = \begin{pmatrix} 40.0 & 40.0 & 40.0 & 40.0 & 40.0 & 40.0 & 40.0 & 71311.0 \end{pmatrix}$$

$$H = \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{pmatrix}$$

$$\mu = 1.0 \cdot 10^{22}$$

$$\alpha = 1.5$$

$$\alpha = \text{diag} \begin{pmatrix} 0.8775 & 0.8081 & 0.6833 \end{pmatrix} \text{ from Equation (4.1)}$$

E.19 Case 1.2.4

$$K_P = \text{diag} \begin{pmatrix} 113.6 & 50.19 & 147.4 \end{pmatrix}$$

$$K_I = \text{diag} \begin{pmatrix} 12.0 & 7.105 & 12.0 \end{pmatrix}$$

$$K_D = \text{diag} \begin{pmatrix} 15.0 & 29.89 & 15.0 \end{pmatrix}$$

$$K_{DD} = \text{diag} \begin{pmatrix} 1.5 & 1.8 & 2.39 \end{pmatrix}$$

$$E_{f1} = \begin{pmatrix} 48733.0 & 35722.0 & 44199.0 & 81688.0 & 10.0 & 16700.0 \end{pmatrix}$$

$$E_{f2} = \begin{pmatrix} 36133.0 & 64122.0 & 4392.0 & 10.0 & 13200.0 & 71611.0 \end{pmatrix}$$

$$E_{f3} = \begin{pmatrix} 49200.0 & 10.0 & 10.0 & 34599.0 & 14633.0 & 10.0 \end{pmatrix}$$

$$E_{f4} = \begin{pmatrix} 50999.0 & 10.0 & 10.0 & 43511.0 & 52577.0 & 57133.0 \end{pmatrix}$$

$$E_{f5} = \begin{pmatrix} 91644.0 & 65600.0 & 99622.0 & 2019.0 & 10.0 & 4786.0 \end{pmatrix}$$

$$E_{g1} = \begin{pmatrix} 1000.0 & 70711.0 & 75799.0 & 1000.0 & 60999.0 & 25299.0 & 59600.0 & 1000.0 \end{pmatrix}$$

$$E_{g2} = \begin{pmatrix} 89766.0 & 19533.0 & 26822.0 & 1000.0 & 1000.0 & 62400.0 & 28588.0 & 1000.0 \end{pmatrix}$$

$$E_{g3} = \begin{pmatrix} 1000.0 & 26499.0 & 2086.0 & 78000.0 & 49444.0 & 94244.0 & 61966.0 & 7910.0 \end{pmatrix}$$

$$E_{g4} = \begin{pmatrix} 69700.0 & 2018.0 & 2405.0 & 39311.0 & 1000.0 & 24044.0 & 1000.0 & 1000.0 \end{pmatrix}$$

$$k = 62.61$$

$$E_{r1} = \begin{pmatrix} 1.0 \cdot 10^{-6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2642 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9089 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.6756 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01217 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.2455 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6762 \end{pmatrix}$$

$$E_{r3} = \begin{pmatrix} 0.04361 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.129 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.09684 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.05672 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9113 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.6134 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.01601 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9286 \end{pmatrix}$$

$$\begin{aligned}
E_{r4} &= \begin{pmatrix} 0.3588 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.173 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.853 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5477 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.859 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7955 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.03959 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1185 \end{pmatrix} \\
E_{r5} &= \begin{pmatrix} 0.326 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5362 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01153 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.733 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1923 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} \end{pmatrix} \\
E_{q1} &= \begin{pmatrix} 24900.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 48266.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 42800.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 61688.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} \end{pmatrix} \\
E_{q2} &= \begin{pmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 37022.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 40300.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 16044.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 39077.0 & 0 \end{pmatrix} \\
E_{q3} &= \begin{pmatrix} 34999.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15477.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 25088.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 72500.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 36966.0 & 0 \end{pmatrix}
\end{aligned}$$

$$E_{q4} = \begin{pmatrix} 79300.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 13399.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 48244.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 95744.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 71477.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 33888.0 \end{pmatrix}$$

$$E_{q5} = \begin{pmatrix} 99211.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 25877.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 18022.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 70322.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 84511.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-6} \end{pmatrix}$$

$$\lambda = 1.0$$

$$modes = \begin{pmatrix} 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 0 & 0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 0 & 0 & 0 & 1.0 & 1.0 & 1.0 & 1.0 \end{pmatrix}$$

$$numberOfModes = 5.0$$

$$p_{ij} = \begin{pmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.5 \end{pmatrix}$$

$$e_{ij} = \begin{pmatrix} 21.36 & 21.36 & 21.36 & 21.36 & 21.36 \\ 21.36 & 21.36 & 21.36 & 21.36 & 21.36 \\ 21.36 & 21.36 & 21.36 & 21.36 & 21.36 \\ 21.36 & 21.36 & 21.36 & 21.36 & 21.36 \\ 21.36 & 21.36 & 21.36 & 21.36 & 21.36 \end{pmatrix}$$

$$\alpha = diag \left(\begin{array}{ccc} 0 & 0.3997 & 0.8558 \end{array} \right) \text{ from Equation (4.1)}$$

E.20 Case 1.2.5.1

$$K_P = diag \left(\begin{array}{ccc} 70.0 & 219.3 & 100.0 \end{array} \right)$$

$$K_I = diag \left(\begin{array}{ccc} 10.0 & 2.128 & 728.9 \end{array} \right)$$

$$K_D = diag \left(\begin{array}{ccc} 40.0 & 40.0 & 70.0 \end{array} \right)$$

$$K_{DD} = diag \left(\begin{array}{ccc} 3.361 & 1.401 & 2.0 \end{array} \right)$$

$$Am = \begin{pmatrix} -3.425 & 0 & 0 \\ 0 & -8.797 & 0 \\ 0 & 0 & -0.02 \end{pmatrix}$$

$$Q = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.0005 \end{pmatrix}$$

$$\gamma_1 = \begin{pmatrix} 0.8 & 0 & 0 \\ 0 & 6.022 & 0 \\ 0 & 0 & 1.969 \end{pmatrix}$$

$$\gamma_2 = \begin{pmatrix} 0.2 & 0 & 0 \\ 0 & 0.2058 & 0 \\ 0 & 0 & 0.4 \end{pmatrix}$$

$$\gamma_3 = \begin{pmatrix} 21.14 & 0 & 0 \\ 0 & 20.3 & 0 \\ 0 & 0 & 17.86 \end{pmatrix}$$

$$\gamma_4 = \begin{pmatrix} 26.66 & 0 & 0 \\ 0 & 27.22 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$\alpha = diag(0 \ 0.9526 \ 0.9201)$ from Equation (4.1)

E.21 Case 1.2.5.2

$$K_P = diag(587.7 \ 367.5 \ 126.3)$$

$$K_I = diag(3.949 \ 7.142 \ 731.3)$$

$$K_D = diag(59.59 \ 40.0 \ 70.0)$$

$$K_{DD} = diag(11.0 \ 8.711 \ 2.394)$$

$$Am = \begin{pmatrix} -2.409 & 0 & 0 & 0 & 0 & 0 \\ 0 & -21.46 & 0 & 0 & 0 & 0 \\ 0 & 0 & -8.418 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.01 \end{pmatrix}$$

$$\begin{aligned}
Q &= \begin{pmatrix} 20.7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6.274 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0005 \end{pmatrix} \\
\gamma_1 &= \begin{pmatrix} 44.6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 22.64 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.968 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0001 \end{pmatrix} \\
\gamma_2 &= \begin{pmatrix} 13.83 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5306 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0001 \end{pmatrix} \\
\gamma_3 &= \begin{pmatrix} 0.001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 18.43 & 0 & 0 & 0 & 0 \\ 0 & 0 & 39.88 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20.53 & 0 & 0 \\ 0 & 0 & 0 & 0 & 14.06 & 0 \\ 0 & 0 & 0 & 0 & 0 & 11.37 \end{pmatrix} \\
\gamma_4 &= \begin{pmatrix} 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 16.26 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 49.08 \end{pmatrix}
\end{aligned}$$

$$\alpha = \text{diag} \left(\begin{array}{ccc} 0 & 0 & 0.9716 \end{array} \right) \text{ from Equation (4.1)}$$

E.22 Case 1.2.5.3

$$K_P = \text{diag} \left(\begin{array}{ccc} 590.4 & 60.0 & 100.0 \end{array} \right)$$

$$K_I = \text{diag} \left(\begin{array}{ccc} 49.63 & 20.0 & 420.6 \end{array} \right)$$

$$K_D = \text{diag} \left(\begin{array}{ccc} 40.0 & 19.56 & 14.96 \end{array} \right)$$

$$K_{DD} = diag \left(\begin{array}{ccc} 18.34 & 15.0 & 2.0 \end{array} \right)$$

$$Am = \begin{pmatrix} -8.968 & 0 & 0 & 0 & 0 & 0 \\ 0 & -40.31 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -31.98 & 0 & 0 \\ 0 & 0 & 0 & 0 & -22.12 & 0 \\ 0 & 0 & 0 & 0 & 0 & -84.46 \end{pmatrix}$$

$$Q = \begin{pmatrix} 32.63 & 0 & 0 & 0 & 0 & 0 \\ 0 & 38.73 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.005 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.005 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0005 \end{pmatrix}$$

$$\gamma_1 = \begin{pmatrix} 2911.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 834.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8823.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1244.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4176.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2135.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 700.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 700.0 \end{pmatrix}$$

$$\gamma_2 = \begin{pmatrix} 7452.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 700.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4602.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9778.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3144.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9674.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3248.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9175.0 \end{pmatrix}$$

$$\gamma_3 = \begin{pmatrix} 9267.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3635.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2748.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1007.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9120.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4120.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5410.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7314.0 \end{pmatrix}$$

$$\gamma_4 = \begin{pmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2250.0 \end{pmatrix}$$

$$B_0 = \begin{pmatrix} -13111.0 & 20000.0 & 3760.0 & -0.05 & 14800.0 & 7129.0 \\ -12799.0 & 19200.0 & -24411.0 & -12677.0 & 3763.0 & 10266.0 \\ 845.5 & 7500.0 & -1044.0 & 8298.0 & 7500.0 & -5770.0 \\ 2934.0 & 15000.0 & 0.05 & 2550.0 & -13711.0 & 21588.0 \\ -7774.0 & 21077.0 & 4236.0 & -274.9 & 0.05 & 20000.0 \\ -4287.0 & 0.05 & -1535.0 & 15000.0 & 7121.0 & -20088.0 \\ 15300.0 & 10700.0 & -558.0 & 0.05 & 20000.0 & -892.4 \\ -23388.0 & -3717.0 & 1813.0 & -3225.0 & -10844.0 & -15000.0 \end{pmatrix}$$

$\alpha = \text{diag} \begin{pmatrix} 0 & 0 & 0.9446 \end{pmatrix}$ from Equation (4.1)

E.23 Case 2.1.1

$$K_P = \begin{pmatrix} 58.84 & 72.95 & 16.53 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 1.511 & 0.04139 & 28.98 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 20.0 & 22.37 & 13.67 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0.7634 & 0.9384 & 0 \end{pmatrix}$$

$$K_{\tau P} = \begin{pmatrix} 352.1 & 401.1 & 123.7 \end{pmatrix}$$

$$K_{\tau I} = \begin{pmatrix} 987.2 & 974.0 & 26.49 \end{pmatrix}$$

$$K_{\tau D} = \begin{pmatrix} 22.23 & 24.58 & 10.0 \end{pmatrix}$$

$\alpha = \text{diag} \begin{pmatrix} 0.9288 & 0.5393 & 0.7102 \end{pmatrix}$ from Equation (4.1)

E.24 Case 2.1.2.1

$$K_P = \begin{pmatrix} 40.0 & 79.44 & 40.0 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 10.0 & 10.0 & 10.0 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 10.0 & 20.57 & 10.0 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

$$c = \begin{pmatrix} 23.39 & 0 & 0 \\ 0 & 25.76 & 0 \\ 0 & 0 & 10.82 \end{pmatrix}$$

$$\lambda = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{pmatrix}$$

$$\alpha = \begin{pmatrix} 2.0 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 2.0 \end{pmatrix}$$

$$\alpha = \text{diag} \left(\begin{array}{ccc} 0.5297 & 0.5684 & 0.9834 \end{array} \right) \text{ from Equation (4.1)}$$

E.25 Case 2.1.2.2

$$K_P = \left(\begin{array}{ccc} 62.89 & 81.46 & 30.0 \end{array} \right)$$

$$K_I = \left(\begin{array}{ccc} 0.9716 & 2.618 & 38.05 \end{array} \right)$$

$$K_D = \left(\begin{array}{ccc} 20.69 & 22.85 & 14.52 \end{array} \right)$$

$$K_{DD} = \left(\begin{array}{ccc} 0 & 0.2086 & 0 \end{array} \right)$$

$$c = \begin{pmatrix} 23.22 & 0 & 0 & 0 & 0 & 0 \\ 0 & 26.42 & 0 & 0 & 0 & 0 \\ 0 & 0 & 11.44 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9.63 & 0 & 0 \\ 0 & 0 & 0 & 0 & 51.63 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6.0 \end{pmatrix}$$

$$\lambda = \begin{pmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 75.07 & 0 & 0 \\ 0 & 0 & 0 & 0 & 68.9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2 \end{pmatrix}$$

$$\alpha = \begin{pmatrix} 0.8761 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2795 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.407 & 0 & 0 & 0 \\ 0 & 0 & 0 & 66.12 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.883 \end{pmatrix}$$

$$\alpha = \text{diag} \left(\begin{array}{ccc} 0.5332 & 0.6038 & 0.9365 \end{array} \right) \text{ from Equation (4.1)}$$

E.26 Case 2.1.2.3

$$K_P = \begin{pmatrix} 45.09 & 66.95 & 40.0 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 1.311 & 0.878 & 58.38 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 10.0 & 20.91 & 10.0 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

$$c = \begin{pmatrix} 29.58 & 0 & 0 & 0 & 0 & 0 \\ 0 & 28.21 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10.27 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.0 \end{pmatrix}$$

$$\lambda = \begin{pmatrix} 33088.0 & -23211.0 & 18044.0 & 44700.0 & 43988.0 & 20122.0 \\ -4000.0 & 4.0 & 4000.0 & -45500.0 & 36777.0 & 4000.0 \\ 400.0 & -4.0 & -4000.0 & -47422.0 & -4.0 & -38499.0 \\ -20422.0 & 1301.0 & -27166.0 & -26200.0 & 13044.0 & -3501.0 \\ -4.0 & -6365.0 & -400.0 & -4.0 & -4000.0 & 400.0 \\ 4000.0 & 20644.0 & -16933.0 & 2641.0 & 4000.0 & -47411.0 \\ 4000.0 & 8175.0 & 4.0 & 4000.0 & -12600.0 & 25711.0 \\ 1172.0 & 4.0 & 4000.0 & 11288.0 & 18177.0 & 29099.0 \end{pmatrix}$$

$$\alpha = \begin{pmatrix} 5500.0 & -17866.0 & 222.2 & 38699.0 & 9901.0 & 46988.0 \\ -5500.0 & 5.5 & 43477.0 & -5500.0 & 42222.0 & 21677.0 \\ 37299.0 & -55.0 & -5500.0 & -550.0 & -40911.0 & -45777.0 \\ -42499.0 & 7318.0 & -10944.0 & -48499.0 & 45022.0 & -36988.0 \\ -49111.0 & -975.5 & -550.0 & -7981.0 & -37322.0 & 550.0 \\ 3376.0 & 23966.0 & -2422.0 & 48077.0 & 5500.0 & -21000.0 \\ 5500.0 & 36600.0 & 55.0 & 5500.0 & -33100.0 & 55.0 \\ 5500.0 & 41099.0 & 5500.0 & 1777.0 & 5.5 & 5500.0 \end{pmatrix}$$

$$\alpha = \text{diag} \begin{pmatrix} 0.4234 & 0.5088 & 0.9445 \end{pmatrix} \text{ from Equation (4.1)}$$

E.27 Case 2.1.3.1

$$K_P = \begin{pmatrix} 40.0 & 58.04 & 20.29 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 2.122 & 10.0 & 23.31 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 10.0 & 10.0 & 10.0 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

$$P = \begin{pmatrix} 7.934 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.659 \cdot 10^6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.574 \cdot 10^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.257 \cdot 10^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8.471 \cdot 10^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{pmatrix}$$

$$Q = \begin{pmatrix} 1.24 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.724 \cdot 10^6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10000.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.87 \cdot 10^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7.816 \cdot 10^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{pmatrix}$$

$$R = \begin{pmatrix} 70.0 & 0 & 0 \\ 0 & 70.0 & 0 \\ 0 & 0 & 7.02 \cdot 10^6 \end{pmatrix}$$

$$E_f = \begin{pmatrix} 0.2 & 0.2 & 0.1 & 73.77 & 232.2 & 0 \end{pmatrix}$$

$$E_g = \begin{pmatrix} 0.02 & 0.02 & 452.7 \end{pmatrix}$$

$$H = \begin{pmatrix} 82.0 \\ 759.1 \\ 859.2 \\ 141.4 \\ 643.8 \\ 498.1 \end{pmatrix}$$

$$\mu = 2.048 \cdot 10^{29}$$

$$\alpha = 1.5$$

$$\alpha = diag \begin{pmatrix} 0.5209 & 0.5599 & 0.8918 \end{pmatrix} \text{ from Equation (4.1)}$$

E.28 Case 2.1.3.2

$$K_P = \begin{pmatrix} 40.0 & 79.42 & 14.83 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 1.195 & 4.057 & 37.17 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 10.0 & 15.47 & 10.0 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

$$\begin{aligned}
P &= \begin{pmatrix} 9.084 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9.831 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.848 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9.262 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4.279 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.502 \cdot 10^8 \end{pmatrix} \\
Q &= \begin{pmatrix} 2.102 \cdot 10^7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.142 \cdot 10^7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.851 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.915 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9.617 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9.479 \cdot 10^8 \end{pmatrix} \\
R &= \begin{pmatrix} 988.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6.809 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 642.6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 180.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} \end{pmatrix} \\
E_f &= \left(\begin{array}{ccccccc} 75599.0 & 24377.0 & 33366.0 & 83400.0 & 18066.0 & 12977.0 \end{array} \right) \\
E_g &= \left(\begin{array}{cccccccc} 9121.0 & 33600.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 \end{array} \right) \\
H &= \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{pmatrix} \\
\mu &= 1.0 \cdot 10^{20} \\
\alpha &= 1.5 \\
\alpha &= \text{diag} \left(\begin{array}{ccc} 0.4935 & 0.5601 & 0.9388 \end{array} \right) \text{ from Equation (4.1)}
\end{aligned}$$

E.29 Case 2.1.3.3

$$K_P = \left(\begin{array}{ccc} 40.0 & 98.48 & 20.74 \end{array} \right)$$

$$K_I = \left(\begin{array}{ccc} 0.0955 & 10.0 & 10.0 \end{array} \right)$$

$$K_D = \begin{pmatrix} 25.53 & 19.76 & 10.0 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0.3758 & 0 \end{pmatrix}$$

$$P = \begin{pmatrix} 1.223 \cdot 10^7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.0 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.999 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.0 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4.059 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.361 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5.0 \cdot 10^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.337 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5.881 \cdot 10^8 \end{pmatrix}$$

$$Q = \begin{pmatrix} 9.175 \cdot 10^7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7.113 \cdot 10^7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.009 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.843 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.668 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8.339 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9.371 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7.654 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.518 \cdot 10^8 \end{pmatrix}$$

$$R = \begin{pmatrix} 303.7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 786.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 \end{pmatrix}$$

$$E_f = \begin{pmatrix} 46477.0 & 38344.0 & 10.0 & 61733.0 & 71577.0 & 10.0 & 11844.0 & 51644.0 & 68733.0 \end{pmatrix}$$

$$E_g = \begin{pmatrix} 40.0 & 40.0 & 40.0 & 18933.0 & 40.0 & 40.0 & 40.0 & 40.0 \end{pmatrix}$$

$$H = \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{pmatrix}$$

$$\mu = 1.0 \cdot 10^{22}$$

$$\alpha = 1.5$$

$\alpha = diag \begin{pmatrix} 0.457 & 0.5004 & 0.9471 \end{pmatrix}$ from Equation (4.1)

E.30 Case 2.1.4

$$K_P = \begin{pmatrix} 40.0 & 62.21 & 11.61 \end{pmatrix}$$

$$K_I = \begin{pmatrix} 1.607 & 3.161 & 45.36 \end{pmatrix}$$

$$K_D = \begin{pmatrix} 17.68 & 15.64 & 10.0 \end{pmatrix}$$

$$K_{DD} = \begin{pmatrix} 0 & 0.1372 & 0 \end{pmatrix}$$

$$modes = \begin{pmatrix} 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 0 & 0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 0 & 0 & 0 & 1.0 & 1.0 & 1.0 & 1.0 \end{pmatrix}$$

$$numberOfModes = 5.0$$

$$P = \begin{pmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \end{pmatrix}$$

$$Q_1 = \begin{pmatrix} 3.724 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8.0 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.255 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.377 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.621 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.979 \cdot 10^8 \end{pmatrix}$$

$$Q_2 = \begin{pmatrix} 3.736 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6.039 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8.587 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.308 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4.336 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.727 \cdot 10^8 \end{pmatrix}$$

$$Q_3 = \begin{pmatrix} 7.348 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9.762 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.0 \cdot 10^5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4.635 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6.654 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.239 \cdot 10^8 \end{pmatrix}$$

$$Q_4 = \begin{pmatrix} 1.857 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7.113 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8.413 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9.314 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5000.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5000.0 \end{pmatrix}$$

$$Q_5 = \begin{pmatrix} 2.094 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.719 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6.179 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.266 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9.819 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7.57 \cdot 10^8 \end{pmatrix}$$

$$R_1 = \begin{pmatrix} 624.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 208.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 665.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 381.7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 456.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 584.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 351.6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 63.79 \end{pmatrix}$$

$$R_2 = \begin{pmatrix} 127.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 745.8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 187.8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 160.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 444.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 657.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 931.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 245.7 \end{pmatrix}$$

$$\begin{aligned}
R_3 &= \left(\begin{array}{ccccccc} 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 725.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 314.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 594.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 44.71 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \end{array} \right) \\
R_4 &= \left(\begin{array}{ccccccc} 317.8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12.29 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 347.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 198.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 456.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 90.17 \\ 0 & 0 & 0 & 0 & 0 & 0 & 303.2 \end{array} \right) \\
R_5 &= \left(\begin{array}{ccccccc} 240.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 331.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 635.7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 369.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 205.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 558.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 395.6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 746.7 \end{array} \right) \\
E_{f1} &= (42322.0 \ 22800.0 \ 15644.0 \ 43400.0 \ 11933.0 \ 56922.0) \\
E_{f2} &= (10000.0 \ 8916.0 \ 75722.0 \ 20144.0 \ 33122.0 \ 33177.0) \\
E_{f3} &= (10000.0 \ 25533.0 \ 7309.0 \ 6614.0 \ 77122.0 \ 10000.0) \\
E_{f4} &= (64744.0 \ 66699.0 \ 48944.0 \ 49622.0 \ 21966.0 \ 8488.0) \\
E_{f5} &= (86233.0 \ 68399.0 \ 5232.0 \ 46066.0 \ 10000.0 \ 54800.0) \\
E_{g1} &= (78977.0 \ 8432.0 \ 68977.0 \ 27400.0 \ 22088.0 \ 85622.0 \ 55611.0 \ 96033.0) \\
E_{g2} &= (23599.0 \ 28733.0 \ 77100.0 \ 73077.0 \ 43522.0 \ 65300.0 \ 65000.0 \ 10677.0) \\
E_{g3} &= (84766.0 \ 67444.0 \ 90500.0 \ 26622.0 \ 83866.0 \ 76222.0 \ 32800.0 \ 1000.0) \\
E_{g4} &= (74199.0 \ 44311.0 \ 65211.0 \ 1000.0 \ 28133.0 \ 76499.0 \ 39200.0 \ 83500.0) \\
E_{g5} &= (63711.0 \ 7213.0 \ 27088.0 \ 83111.0 \ 13533.0 \ 84766.0 \ 29066.0 \ 53177.0)
\end{aligned}$$

$$H_1 = \begin{pmatrix} 84033.0 \\ 87233.0 \\ 42133.0 \\ 1.0 \\ 80388.0 \\ 18811.0 \end{pmatrix}$$

$$H_2 = \begin{pmatrix} 22944.0 \\ 1.0 \\ 38744.0 \\ 94266.0 \\ 27988.0 \\ 76344.0 \end{pmatrix}$$

$$H_3 = \begin{pmatrix} 48200.0 \\ 1.0 \\ 59200.0 \\ 39211.0 \\ 30566.0 \\ 45211.0 \end{pmatrix}$$

$$H_4 = \begin{pmatrix} 70833.0 \\ 58311.0 \\ 44177.0 \\ 53477.0 \\ 2503.0 \\ 26377.0 \end{pmatrix}$$

$$H_5 = \begin{pmatrix} 44266.0 \\ 68044.0 \\ 32300.0 \\ 78688.0 \\ 59844.0 \\ 32800.0 \end{pmatrix}$$

$$p_{ij} = \begin{pmatrix} 21.75 & 0 & 0 & 0 & 0 \\ 0 & 21.75 & 0 & 0 & 0 \\ 0 & 0 & 21.75 & 0 & 0 \\ 0 & 0 & 0 & 21.75 & 0 \\ 0 & 0 & 0 & 0 & 21.75 \end{pmatrix}$$

$$e_i = (50.39 \ 50.39 \ 50.39 \ 50.39 \ 50.39)$$

$$k = 224.9$$

$$\mu = 5.722 \cdot 10^{29}$$

$$\alpha = 49.39$$

$$\alpha = \text{diag} \begin{pmatrix} 0.428 & 0.486 & 0.9741 \end{pmatrix} \text{ from Equation (4.1)}$$

E.31 Case 2.2.1

$$K_P = \text{diag} \begin{pmatrix} 90.0 & 90.0 & 90.0 \end{pmatrix}$$

$$K_I = \text{diag} \begin{pmatrix} 10.0 & 10.0 & 21.37 \end{pmatrix}$$

$$K_D = \text{diag} \begin{pmatrix} 40.0 & 40.0 & 17.8 \end{pmatrix}$$

$$K_{DD} = \text{diag} \begin{pmatrix} 1.114 & 2.0 & 2.0 \end{pmatrix}$$

$$K_{\tau P} = \text{diag} \begin{pmatrix} 115.4 & 120.7 & 25.95 \end{pmatrix}$$

$$K_{\tau I} = \text{diag} \begin{pmatrix} 6.808 & 50.42 & 14.77 \end{pmatrix}$$

$$K_{\tau D} = \text{diag} \begin{pmatrix} 14.0 & 14.0 & 2.482 \end{pmatrix}$$

$$\alpha = \text{diag} \begin{pmatrix} 0.111 & 0.106 & 0.5 \end{pmatrix}$$

from Equation (4.1)

E.32 Case 2.2.2.1

$$K_P = \text{diag} \begin{pmatrix} 175.0 & 122.8 & 100.0 \end{pmatrix}$$

$$K_I = \text{diag} \begin{pmatrix} 30.51 & 0.7409 & 161.2 \end{pmatrix}$$

$$K_D = \text{diag} \begin{pmatrix} 33.61 & 32.84 & 31.6 \end{pmatrix}$$

$$K_{DD} = \text{diag} \begin{pmatrix} 13.92 & 11.91 & 2.196 \end{pmatrix}$$

$$c = \begin{pmatrix} 16.8 & 0 & 0 \\ 0 & 18.26 & 0 \\ 0 & 0 & 3.0 \end{pmatrix}$$

$$\lambda = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{pmatrix}$$

$$\alpha = \begin{pmatrix} 0.1493 & 0 & 0 \\ 0 & 0.3489 & 0 \\ 0 & 0 & 2.861 \end{pmatrix}$$

$$\alpha = \text{diag} \begin{pmatrix} 0.9342 & 0.938 & 0.9543 \end{pmatrix} \text{ from Equation (4.1)}$$

E.33 Case 2.2.2.2

$$K_P = \text{diag} \begin{pmatrix} 90.68 & 72.54 & 100.0 \end{pmatrix}$$

$$K_I = \text{diag} \begin{pmatrix} 46.84 & 10.0 & 56.31 \end{pmatrix}$$

$$K_D = \text{diag} \begin{pmatrix} 25.56 & 22.41 & 35.18 \end{pmatrix}$$

$$K_{DD} = \text{diag} \begin{pmatrix} 2.878 & 0.4223 & 7.595 \end{pmatrix}$$

$$c = \begin{pmatrix} 0.2763 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1697 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 69.39 & 0 & 0 \\ 0 & 0 & 0 & 0 & 45.21 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6.0 \end{pmatrix}$$

$$\lambda = \begin{pmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 33.64 & 0 & 0 \\ 0 & 0 & 0 & 0 & 54.16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2 \end{pmatrix}$$

$$\alpha = \begin{pmatrix} 2.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.199 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7.615 & 0 & 0 & 0 \\ 0 & 0 & 0 & 35.83 & 0 & 0 \\ 0 & 0 & 0 & 0 & 80.89 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.009762 \end{pmatrix}$$

$$\alpha = \text{diag} \begin{pmatrix} 0.008067 & 0 & 0.9753 \end{pmatrix} \text{ from Equation (4.1)}$$

E.34 Case 2.2.2.3

$$K_P = \text{diag} \begin{pmatrix} 175.0 & 167.2 & 100.0 \end{pmatrix}$$

$$K_I = \text{diag} \begin{pmatrix} 30.51 & 10.0 & 56.31 \end{pmatrix}$$

$$K_D = \text{diag} \begin{pmatrix} 45.54 & 30.79 & 37.21 \end{pmatrix}$$

$$K_{DD} = \text{diag} \begin{pmatrix} 11.75 & 10.94 & 9.658 \end{pmatrix}$$

$$\begin{aligned}
c &= \begin{pmatrix} 33.19 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7.878 \end{pmatrix} \\
\lambda &= \begin{pmatrix} 1000.0 & -100.0 & 20800.0 & 1000.0 & 13666.0 & 1.0 \\ -1000.0 & 1.0 & 1000.0 & -31177.0 & 1.0 & 44800.0 \\ 100.0 & -918.4 & -40766.0 & -33777.0 & -30511.0 & -1000.0 \\ -560.9 & 1000.0 & -23300.0 & -1.0 & 44799.0 & -29611.0 \\ -1.0 & -511.2 & -9753.0 & -1.0 & -29122.0 & 47866.0 \\ 1000.0 & 1000.0 & -20677.0 & 20033.0 & 44644.0 & -1434.0 \\ 183.6 & 100.0 & 1.0 & 3459.0 & -100.0 & 1.0 \\ 1000.0 & 1.0 & 39866.0 & 35888.0 & 31900.0 & 42199.0 \end{pmatrix} \\
\alpha &= \begin{pmatrix} 24033.0 & -34922.0 & 55.0 & 5500.0 & 42822.0 & 55.0 \\ -5242.0 & 2078.0 & 5500.0 & -48911.0 & 39699.0 & 5500.0 \\ 36099.0 & -4541.0 & -36888.0 & -30188.0 & -55.0 & -48599.0 \\ -24588.0 & 2983.0 & -24888.0 & -28544.0 & 21133.0 & -31900.0 \\ -55.0 & -5500.0 & -911.5 & -12966.0 & -10811.0 & 27400.0 \\ 35866.0 & 43700.0 & -5384.0 & 6050.0 & 14500.0 & -5.5 \\ 5300.0 & 17244.0 & 55.0 & 5500.0 & -18877.0 & 55.0 \\ 5500.0 & 1779.0 & 5500.0 & 5500.0 & 33722.0 & 34499.0 \end{pmatrix}
\end{aligned}$$

$$\alpha = \text{diag} \begin{pmatrix} 0.8905 & 0 & 0.9904 \end{pmatrix} \text{ from Equation (4.1)}$$

E.35 Case 2.2.3.1

$$K_P = \text{diag} \begin{pmatrix} 108.0 & 0.7873 & 83.6 \end{pmatrix}$$

$$K_I = \text{diag} \begin{pmatrix} 2.355 & 36.95 & 124.9 \end{pmatrix}$$

$$K_D = \text{diag} \begin{pmatrix} 4.382 & 0.3149 & 44.85 \end{pmatrix}$$

$$K_{DD} = \text{diag} \begin{pmatrix} 2.762 & 2.622 & 5.975 \end{pmatrix}$$

$$P = \begin{pmatrix} 1.531 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6.988 \cdot 10^6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8.323 \cdot 10^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4.298 \cdot 10^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.93 \cdot 10^5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8.057 \cdot 10^6 \end{pmatrix}$$

$$Q = \begin{pmatrix} 47822.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.747 \cdot 10^6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.001 \cdot 10^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6.587 \cdot 10^5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7.962 \cdot 10^6 \end{pmatrix}$$

$$R = \begin{pmatrix} 1.09 \cdot 10^6 & 0 & 0 \\ 0 & 4.216 \cdot 10^5 & 0 \\ 0 & 0 & 9.174 \cdot 10^6 \end{pmatrix}$$

$$E_f = \begin{pmatrix} 131.3 & 0.2 & 15.52 & 718.0 & 0 & 734.6 \end{pmatrix}$$

$$E_g = \begin{pmatrix} 0.02 & 954.4 & 0.02 \end{pmatrix}$$

$$H = \begin{pmatrix} 63.11 \\ 398.9 \\ 441.1 \\ 185.3 \\ 417.5 \\ 643.6 \end{pmatrix}$$

$$\mu = 1.559 \cdot 10^{29}$$

$$\text{alpha} = 1.5$$

$$\alpha = \text{diag} \begin{pmatrix} 0.9652 & 0 & 0.96 \end{pmatrix} \text{ from Equation (4.1)}$$

E.36 Case 2.2.3.2

$$K_P = \text{diag} \begin{pmatrix} 70.0 & 70.0 & 60.0 \end{pmatrix}$$

$$K_I = \text{diag} \begin{pmatrix} 2.0 & 2.0 & 105.8 \end{pmatrix}$$

$$K_D = \text{diag} \begin{pmatrix} 25.0 & 25.0 & 30.0 \end{pmatrix}$$

$$K_{DD} = \text{diag} \begin{pmatrix} 3.0 & 3.0 & 3.0 \end{pmatrix}$$

$$P = \begin{pmatrix} 5.0 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8.93 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7.234 \cdot 10^7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.008 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9.655 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.0 \cdot 10^6 \end{pmatrix}$$

$$Q = \begin{pmatrix} 3.721 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.604 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.0 \cdot 10^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.0 \cdot 10^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.307 \cdot 10^7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.94 \cdot 10^7 \end{pmatrix}$$

$$R = \begin{pmatrix} 953.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 840.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 39.99 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 351.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} \end{pmatrix}$$

$$E_f = \begin{pmatrix} 20.0 & 20.0 & 10.0 & 52911.0 & 48499.0 & 23177.0 \end{pmatrix}$$

$$E_g = \begin{pmatrix} 67944.0 & 42122.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 & 1000.0 \end{pmatrix}$$

$$H = \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{pmatrix}$$

$$\mu = 1.0 \cdot 10^{20}$$

$$\alpha = 1.5$$

$$\alpha = \text{diag} \begin{pmatrix} 0.8822 & 0.8796 & 0.9528 \end{pmatrix} \text{ from Equation (4.1)}$$

E.37 Case 2.2.3.3

$$K_P = \text{diag} \begin{pmatrix} 70.0 & 103.5 & 66.26 \end{pmatrix}$$

$$K_I = \text{diag} \begin{pmatrix} 5.0 & 5.0 & 45.29 \end{pmatrix}$$

$$K_D = \text{diag} \begin{pmatrix} 31.04 & 40.0 & 26.25 \end{pmatrix}$$

$$K_{DD} = \text{diag} \begin{pmatrix} 5.319 & 6.394 & 1.0 \end{pmatrix}$$

$$P = \begin{pmatrix} 6.67 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7.993 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.799 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6.561 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.606 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.067 \cdot 10^7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3.67 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.765 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5.0 \cdot 10^6 \end{pmatrix}$$

$$Q = \begin{pmatrix} 5.0 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.915 \cdot 10^7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.0 \cdot 10^6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8.188 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8.052 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.0 \cdot 10^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5.0 \cdot 10^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.253 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.746 \cdot 10^7 \end{pmatrix}$$

$$R = \begin{pmatrix} 961.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8.677 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 132.6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \cdot 10^{-5} \end{pmatrix}$$

$$E_f = \left(\begin{array}{cccccccccc} 64477.0 & 86000.0 & 61300.0 & 4863.0 & 34344.0 & 5291.0 & 7479.0 & 0 & 64011.0 \end{array} \right)$$

$$E_g = \left(\begin{array}{cccccccccc} 62033.0 & 58100.0 & 40.0 & 40.0 & 40.0 & 40.0 & 65944.0 & 40.0 \end{array} \right)$$

$$H = \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{pmatrix}$$

$$\mu = 1.0 \cdot 10^{22}$$

$$\alpha = 1.5$$

$$\alpha = \text{diag} \begin{pmatrix} 0.8883 & 0.9218 & 0.951 \end{pmatrix}$$

from Equation (4.1)

E.38 Case 2.2.4

$$K_P = \text{diag} \begin{pmatrix} 40.0 & 1.605 & 161.2 \end{pmatrix}$$

$$K_I = \text{diag} \begin{pmatrix} 16.9 & 0.05122 & 117.4 \end{pmatrix}$$

$$K_D = \text{diag} \begin{pmatrix} 15.0 & 2.101 & 73.97 \end{pmatrix}$$

$$K_{DD} = \text{diag} \begin{pmatrix} 1.5 & 0.5387 & 4.323 \end{pmatrix}$$

$$modes = \begin{pmatrix} 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 0 & 0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 0 & 0 & 0 & 1.0 & 1.0 & 1.0 & 1.0 \end{pmatrix}$$

numberOfModes = 5.0

$$P = \begin{pmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \end{pmatrix}$$

$$Q_1 = \begin{pmatrix} 9.787 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.574 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.053 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.529 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.258 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.371 \cdot 10^7 \end{pmatrix}$$

$$Q_2 = \begin{pmatrix} 8.317 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6.866 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8.642 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.16 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6.157 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.258 \cdot 10^8 \end{pmatrix}$$

$$\begin{aligned}
Q_3 &= \begin{pmatrix} 1.0 \cdot 10^9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 \cdot 10^9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8.847 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8.107 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7.435 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.895 \cdot 10^8 \end{pmatrix} \\
Q_4 &= \begin{pmatrix} 4.686 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.829 \cdot 10^7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7.998 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5000.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4.288 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8.183 \cdot 10^8 \end{pmatrix} \\
Q_5 &= \begin{pmatrix} 3.8 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8.951 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.119 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.489 \cdot 10^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.231 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.308 \cdot 10^8 \end{pmatrix} \\
R_1 &= \begin{pmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 556.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \end{pmatrix} \\
R_2 &= \begin{pmatrix} 686.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 160.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 383.6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 946.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 874.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 37.88 & 0 \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
R_3 &= \left(\begin{array}{ccccccc} 109.7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 268.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 242.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 617.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 344.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 850.0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 707.8 \end{array} \right) \\
R_4 &= \left(\begin{array}{ccccccc} 507.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 877.4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 400.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 607.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 290.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 687.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 144.9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 506.5 \end{array} \right) \\
R_5 &= \left(\begin{array}{ccccccc} 577.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 34.33 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 265.7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 117.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 825.3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 617.0 \end{array} \right) \\
E_{f1} &= (45811.0 \ 91800.0 \ 2544.0 \ 10000.0 \ 23499.0 \ 10000.0) \\
E_{f2} &= (42333.0 \ 8752.0 \ 39100.0 \ 10000.0 \ 79499.0 \ 73188.0) \\
E_{f3} &= (44111.0 \ 10400.0 \ 38511.0 \ 21800.0 \ 47633.0 \ 10000.0) \\
E_{f4} &= (5734.0 \ 99122.0 \ 64111.0 \ 10000.0 \ 51600.0 \ 74200.0) \\
E_{f5} &= (84988.0 \ 76644.0 \ 7750.0 \ 49900.0 \ 94077.0 \ 5584.0) \\
E_{g1} &= (69499.0 \ 67477.0 \ 43788.0 \ 31499.0 \ 77300.0 \ 18444.0 \ 72244.0 \ 72722.0) \\
E_{g2} &= (94100.0 \ 65177.0 \ 77799.0 \ 68733.0 \ 34100.0 \ 52544.0 \ 11988.0 \ 67399.0) \\
E_{g3} &= (88433.0 \ 88400.0 \ 25600.0 \ 33222.0 \ 22888.0 \ 67022.0 \ 149.7 \ 71422.0) \\
E_{g4} &= (69022.0 \ 27044.0 \ 85744.0 \ 40488.0 \ 22822.0 \ 93244.0 \ 55499.0 \ 33322.0) \\
E_{g5} &= (80800.0 \ 59333.0 \ 1000.0 \ 84433.0 \ 91422.0 \ 62422.0 \ 96911.0 \ 14400.0)
\end{aligned}$$

$$\begin{aligned}
H_1 &= \begin{pmatrix} 96000.0 \\ 72066.0 \\ 18433.0 \\ 68788.0 \\ 19300.0 \\ 26500.0 \end{pmatrix} \\
H_2 &= \begin{pmatrix} 9367.0 \\ 48488.0 \\ 87622.0 \\ 47033.0 \\ 94511.0 \\ 23533.0 \end{pmatrix} \\
H_3 &= \begin{pmatrix} 10444.0 \\ 46988.0 \\ 79522.0 \\ 75600.0 \\ 85577.0 \\ 52988.0 \end{pmatrix} \\
H_4 &= \begin{pmatrix} 87144.0 \\ 17199.0 \\ 17933.0 \\ 8801.0 \\ 78444.0 \\ 38622.0 \end{pmatrix} \\
H_5 &= \begin{pmatrix} 68177.0 \\ 21211.0 \\ 38344.0 \\ 81622.0 \\ 71500.0 \\ 28700.0 \end{pmatrix} \\
p_{ij} &= \begin{pmatrix} 99.3 & 0 & 0 & 0 & 0 \\ 0 & 99.3 & 0 & 0 & 0 \\ 0 & 0 & 99.3 & 0 & 0 \\ 0 & 0 & 0 & 99.3 & 0 \\ 0 & 0 & 0 & 0 & 99.3 \end{pmatrix}
\end{aligned}$$

$$e_i = (36.31 \ 36.31 \ 36.31 \ 36.31 \ 36.31)$$

$$k = 238.3$$

$$\mu = 1.0 \cdot 10^{10}$$

$$\alpha = 1.5$$

$\alpha = diag \begin{pmatrix} 0.9468 & 0.8924 & 0.831 \end{pmatrix}$ from Equation (4.1)

APPENDIX F – SENSITIVITY INDICES FOR MODEL 1 PASSIVE AND ACTIVE CASES

Table 36 presents Total effect and Sensitivity indices for simulation success of Passive cases, while Tables 37 and 38 present the same for position error and RMS power, respectively.

Table 36: System stability robustness metrics for Model 1 Passive Cases

Case	$\hat{S}_{T1,p_{sim}}$	$\hat{S}_{T2,p_{sim}}$	$\hat{S}_{T3,p_{sim}}$	$\hat{S}_{T4,p_{sim}}$	$\hat{S}_{T5,p_{sim}}$	$\hat{S}_{T6,p_{sim}}$	$\hat{S}_{T7,p_{sim}}$	$\hat{S}_{1,p_{sim}}$	$\hat{S}_{2,p_{sim}}$	$\hat{S}_{3,p_{sim}}$	$\hat{S}_{4,p_{sim}}$	$\hat{S}_{5,p_{sim}}$	$\hat{S}_{6,p_{sim}}$	$\hat{S}_{7,p_{sim}}$
1.1.1	0.12	0.059	0.073	0.31	0.14	0.73	0.43	0.13	0.024	0.038	0.13	0.14	0.13	-0.025
1.1.2.1	0.028	0.11	0.015	0.22	0.027	0.84	0.1	0.046	0.022	$9.2 \cdot 10^{-3}$	0.11	0.035	0.44	0.052
1.1.2.2	0.051	0.039	$-7.6 \cdot 10^{-3}$	0.25	0.01	0.81	0.26	0.066	$-2.8 \cdot 10^{-3}$	0.022	0.11	$-3.6 \cdot 10^{-3}$	0.33	0.098
1.1.2.3	0.14	0.18	0.14	0.14	0.046	0.82	0.32	$-3.5 \cdot 10^{-3}$	$-7.1 \cdot 10^{-3}$	0.062	$1.7 \cdot 10^{-3}$	0.023	0.35	0.027
1.1.3.1	0.088	0.074	0.059	0.39	0.019	0.66	0.21	0.085	$-9.8 \cdot 10^{-3}$	0.029	0.21	0.031	0.33	0.068
1.1.3.2	0.34	0.18	0.21	0.49	0.12	0.37	0.43	0.2	-0.016	0.034	0.15	0.063	0.088	-0.036
1.1.3.3	0.22	0.12	0.21	0.23	0.093	0.62	0.48	0.15	0.044	0.039	0.11	0.077	0.11	0.14
1.1.4	0.15	0.23	0.099	0.53	0.04	0.38	0.32	0.15	0.042	0.043	0.28	$7.9 \cdot 10^{-3}$	0.11	0.016
1.1.5.1	0.082	0.11	0.096	0.05	0.023	1.0	0.065	-0.012	0.016	0.015	$9.2 \cdot 10^{-3}$	$1.5 \cdot 10^{-3}$	0.75	$-1.3 \cdot 10^{-3}$
1.1.5.2	0.23	0.44	0.32	0.2	0.15	0.87	0.17	0.025	0.077	0.037	$4.6 \cdot 10^{-3}$	$7.7 \cdot 10^{-4}$	0.44	$-4.0 \cdot 10^{-3}$
1.1.5.3	0.16	0.86	0.15	0.21	0.15	0.11	0.18	0.031	0.85	0.016	0.011	0.016	0.15	0.013

Table 37: Position error robustness metrics for Model 1 Passive Cases

Case	$\hat{S}_{T1,pos}$	$\hat{S}_{T2,pos}$	$\hat{S}_{T3,pos}$	$\hat{S}_{T4,pos}$	$\hat{S}_{T5,pos}$	$\hat{S}_{T6,pos}$	$\hat{S}_{T7,pos}$	$\hat{S}_{1,pos}$	$\hat{S}_{2,pos}$	$\hat{S}_{3,pos}$	$\hat{S}_{4,pos}$	$\hat{S}_{5,pos}$	$\hat{S}_{6,pos}$	$\hat{S}_{7,pos}$
1.1.1	0.32	0.2	0.15	0.29	0.31	0.63	0.39	$-8.2 \cdot 10^{-3}$	0.048	0.027	0.055	0.13	0.28	0.1
1.1.2.1	0.23	0.27	0.14	0.32	0.19	0.71	0.22	-0.013	0.1	0.02	0.057	0.038	0.52	0.056
1.1.2.2	0.26	0.16	0.12	0.28	0.15	0.66	0.3	$-5.2 \cdot 10^{-3}$	0.088	$9.6 \cdot 10^{-3}$	0.087	0.033	0.48	0.091
1.1.2.3	0.27	0.32	0.21	0.2	0.24	0.67	0.3	$6.0 \cdot 10^{-3}$	0.18	0.031	0.037	0.059	0.43	0.11
1.1.3.1	0.27	0.23	0.17	0.38	0.2	0.69	0.27	$1.8 \cdot 10^{-4}$	0.054	0.017	0.16	0.018	0.4	0.07
1.1.3.2	0.42	0.17	0.12	0.23	0.064	0.36	0.24	0.1	0.043	0.017	0.11	0.077	0.11	0.096
1.1.3.3	0.4	0.26	0.24	0.29	0.3	0.55	0.48	0.026	0.12	0.054	0.026	0.073	0.24	0.15
1.1.4	0.34	0.37	0.1	0.35	0.15	0.4	0.23	-0.022	0.19	0.04	0.19	0.036	0.17	0.1
1.1.5.1	0.49	0.58	0.01	0.4	-0.039	0.83	0.4	-0.051	0.021	$5.2 \cdot 10^{-3}$	0.14	$1.3 \cdot 10^{-3}$	0.31	$2.5 \cdot 10^{-3}$
1.1.5.2	0.48	0.8	0.057	0.5	-0.11	0.89	0.42	-0.024	-0.027	-0.022	0.12	0.01	$-7.2 \cdot 10^{-3}$	-0.045
1.1.5.3	0.45	0.29	0.04	0.44	0.04	0.71	0.68	-0.025	$7.9 \cdot 10^{-3}$	$8.7 \cdot 10^{-4}$	0.096	$8.9 \cdot 10^{-4}$	0.11	0.012

Table 38: RMS Power robustness metrics for Model 1 Passive Cases

Case	$\hat{S}_{T1,pow}$	$\hat{S}_{T2,pow}$	$\hat{S}_{T3,pow}$	$\hat{S}_{T4,pow}$	$\hat{S}_{T5,pow}$	$\hat{S}_{T6,pow}$	$\hat{S}_{T7,pow}$	$\hat{S}_{1,pow}$	$\hat{S}_{2,pow}$	$\hat{S}_{3,pow}$	$\hat{S}_{4,pow}$	$\hat{S}_{5,pow}$	$\hat{S}_{6,pow}$	$\hat{S}_{7,pow}$
1.1.1	0.33	0.19	0.14	0.28	0.1	0.81	0.55	-0.036	$7.4 \cdot 10^{-3}$	0.022	-0.027	0.067	0.31	0.12
1.1.2.1	0.22	0.25	0.12	0.27	0.18	0.75	0.27	-0.015	0.13	0.028	$-4.2 \cdot 10^{-3}$	0.02	0.53	0.086
1.1.2.2	0.22	0.039	0.03	0.22	$-9.3 \cdot 10^{-3}$	0.74	0.34	-0.061	0.083	$-1.7 \cdot 10^{-3}$	0.017	0.026	0.51	0.11
1.1.2.3	0.26	0.33	0.17	0.22	0.14	0.73	0.33	-0.018	0.16	0.02	0.052	0.031	0.41	0.084
1.1.3.1	0.34	0.33	0.15	0.33	0.13	0.71	0.36	$1.7 \cdot 10^{-3}$	0.096	0.078	0.038	0.037	0.27	0.21
1.1.3.2	0.47	0.34	0.26	0.26	0.098	0.47	0.44	0.039	0.045	0.03	0.091	0.094	0.12	0.21
1.1.3.3	0.4	0.21	0.24	0.29	0.13	0.64	0.58	-0.02	0.12	0.023	$-5.7 \cdot 10^{-3}$	0.041	0.2	0.19
1.1.4	0.44	0.47	0.23	0.32	0.18	0.53	0.3	-0.066	0.26	0.019	0.12	0.082	0.059	0.18
1.1.5.1	0.21	0.27	0.14	0.38	0.13	0.68	0.21	0.038	0.087	$6.7 \cdot 10^{-3}$	0.17	$3.8 \cdot 10^{-4}$	0.62	$4.7 \cdot 10^{-3}$
1.1.5.2	0.28	0.53	0.11	0.31	0.075	0.8	0.35	0.093	0.054	0.014	0.12	$7.0 \cdot 10^{-3}$	0.42	-0.013
1.1.5.3	0.17	0.15	-0.025	0.28	-0.025	0.7	0.34	0.025	$-1.6 \cdot 10^{-3}$	$2.2 \cdot 10^{-4}$	0.12	$2.8 \cdot 10^{-4}$	0.33	0.053

Table 39 presents Total effect and Sensitivity indices for simulation success of Active cases, while Tables 40 and 41 present the same for position error and RMS power, respectively.

Table 39: System stability robustness metrics for Model 1 Active Cases

Case	$\hat{S}_{T1,p_{sim}}$	$\hat{S}_{T2,p_{sim}}$	$\hat{S}_{T3,p_{sim}}$	$\hat{S}_{T4,p_{sim}}$	$\hat{S}_{T5,p_{sim}}$	$\hat{S}_{T6,p_{sim}}$	$\hat{S}_{T7,p_{sim}}$	$\hat{S}_{1,p_{sim}}$	$\hat{S}_{2,p_{sim}}$	$\hat{S}_{3,p_{sim}}$	$\hat{S}_{4,p_{sim}}$	$\hat{S}_{5,p_{sim}}$	$\hat{S}_{6,p_{sim}}$	$\hat{S}_{7,p_{sim}}$
2.1.1	0.13	0.043	0.066	0.56	0.59	0.1	-0.039	0.078	-0.035	0.018	0.19	0.44	$-7.5 \cdot 10^{-3}$	0.077
2.1.2.1	0.24	0.12	0.15	0.48	0.51	0.089	0.046	0.053	0.026	0.04	0.29	0.53	0.068	0.085
2.1.2.2	0.22	0.045	0.17	0.48	0.55	0.2	0.18	0.042	0.078	$4.7 \cdot 10^{-4}$	0.22	0.49	-0.029	-0.021
2.1.2.3	0.2	0.13	0.12	0.46	0.36	0.48	0.17	-0.045	-0.1	-0.063	0.12	0.32	0.033	-0.1
2.1.3.1	0.17	0.075	0.11	0.67	0.42	0.12	-0.047	0.054	-0.027	$4.6 \cdot 10^{-3}$	0.27	0.46	-0.051	0.058
2.1.3.2	0.11	0.042	0.075	0.55	0.49	0.092	-0.021	0.086	-0.022	0.018	0.2	0.57	-0.07	0.015
2.1.3.3	0.17	0.072	0.23	0.59	0.51	0.26	0.15	0.046	0.01	$-7.6 \cdot 10^{-3}$	0.19	0.42	-0.061	-0.041
2.1.4	0.11	0.021	0.064	0.53	0.34	0.25	0.082	0.12	0.05	0.11	0.33	0.51	-0.051	0.012

Table 40: Position error robustness metrics for Model 1 Active Cases

Case	$\hat{S}_{T1,pos}$	$\hat{S}_{T2,pos}$	$\hat{S}_{T3,pos}$	$\hat{S}_{T4,pos}$	$\hat{S}_{T5,pos}$	$\hat{S}_{T6,pos}$	$\hat{S}_{T7,pos}$	$\hat{S}_{1,pos}$	$\hat{S}_{2,pos}$	$\hat{S}_{3,pos}$	$\hat{S}_{4,pos}$	$\hat{S}_{5,pos}$	$\hat{S}_{6,pos}$	$\hat{S}_{7,pos}$
2.1.1	0.34	0.21	0.19	0.46	0.79	0.24	0.21	$2.9 \cdot 10^{-3}$	0.029	0.027	0.12	0.46	0.021	-0.012
2.1.2.1	0.31	0.27	0.21	0.53	0.75	0.18	0.2	0.034	0.048	0.03	0.086	0.46	0.032	$9.0 \cdot 10^{-5}$
2.1.2.2	0.35	0.35	0.24	0.51	0.74	0.32	0.31	0.036	0.064	0.059	0.11	0.35	0.049	0.035
2.1.2.3	0.39	0.38	0.3	0.48	0.63	0.57	0.31	0.037	0.073	0.046	0.042	0.22	0.23	0.028
2.1.3.1	0.32	0.33	0.21	0.56	0.71	0.23	0.18	0.043	0.075	0.029	0.2	0.26	0.047	-0.01
2.1.3.2	0.38	0.31	0.25	0.55	0.79	0.2	0.21	0.014	0.028	0.021	0.11	0.35	0.039	$5.5 \cdot 10^{-3}$
2.1.3.3	0.34	0.35	0.3	0.53	0.73	0.33	0.26	0.024	0.049	0.057	0.13	0.31	0.052	0.022
2.1.4	0.39	0.29	0.27	0.58	0.71	0.31	0.26	0.019	0.021	$9.1 \cdot 10^{-3}$	0.092	0.28	0.048	-0.014

Table 41: RMS Power robustness metrics for Model 1 Active Cases

Case	$\hat{S}_{T1,pow}$	$\hat{S}_{T2,pow}$	$\hat{S}_{T3,pow}$	$\hat{S}_{T4,pow}$	$\hat{S}_{T5,pow}$	$\hat{S}_{T6,pow}$	$\hat{S}_{T7,pow}$	$\hat{S}_{1,pow}$	$\hat{S}_{2,pow}$	$\hat{S}_{3,pow}$	$\hat{S}_{4,pow}$	$\hat{S}_{5,pow}$	$\hat{S}_{6,pow}$	$\hat{S}_{7,pow}$
2.1.1	0.5	0.21	0.18	0.36	0.75	0.27	0.27	0.11	0.029	0.012	0.045	0.41	0.016	-0.019
2.1.2.1	0.4	0.31	0.28	0.52	0.72	0.3	0.3	0.12	$-5.0 \cdot 10^{-3}$	-0.019	-0.025	0.48	-0.011	-0.035
2.1.2.2	0.34	0.33	0.19	0.41	0.68	0.34	0.38	0.12	0.075	0.09	0.036	0.35	0.043	0.081
2.1.2.3	0.52	0.43	0.38	0.49	0.62	0.67	0.41	0.052	0.053	0.02	$8.7 \cdot 10^{-3}$	0.12	0.2	0.04
2.1.3.1	0.46	0.27	0.25	0.43	0.63	0.2	0.22	0.28	0.054	$-4.6 \cdot 10^{-4}$	0.15	0.16	0.039	$5.3 \cdot 10^{-3}$
2.1.3.2	0.51	0.31	0.35	0.49	0.74	0.26	0.29	0.21	0.036	-0.025	0.05	0.25	0.047	$8.3 \cdot 10^{-3}$
2.1.3.3	0.51	0.42	0.4	0.52	0.76	0.47	0.39	0.11	0.029	0.056	0.061	0.26	0.053	0.048
2.1.4	0.52	0.25	0.29	0.52	0.55	0.29	0.31	0.14	$-7.4 \cdot 10^{-3}$	-0.038	0.16	0.13	0.01	-0.011

APPENDIX G – FDD ROBUSTNESS GRAPHS FOR MODEL 2

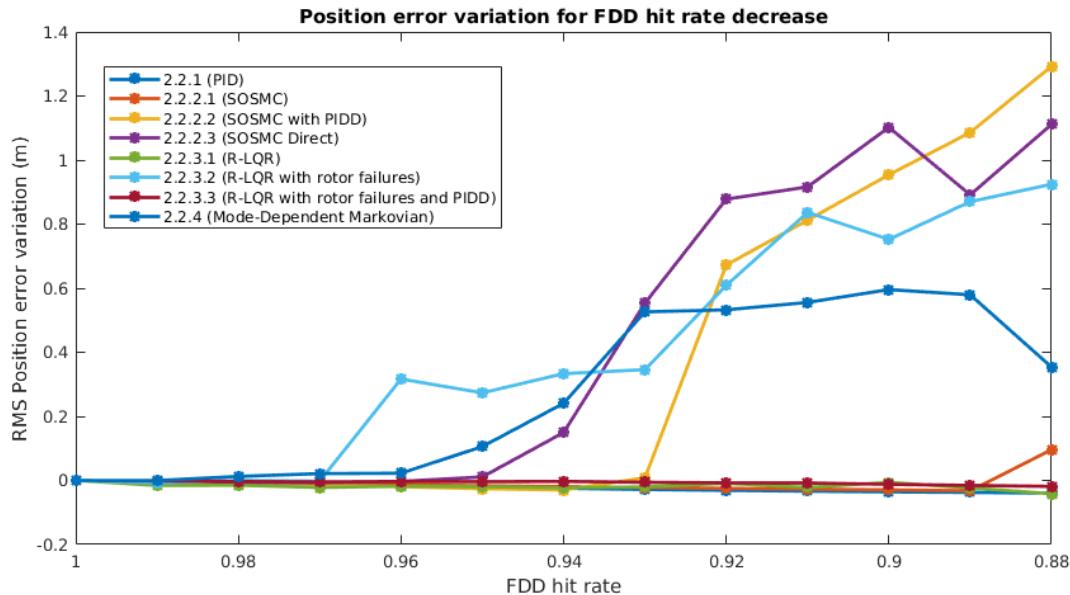


Figure 55: Position error variation to FDD hit rate decrease.

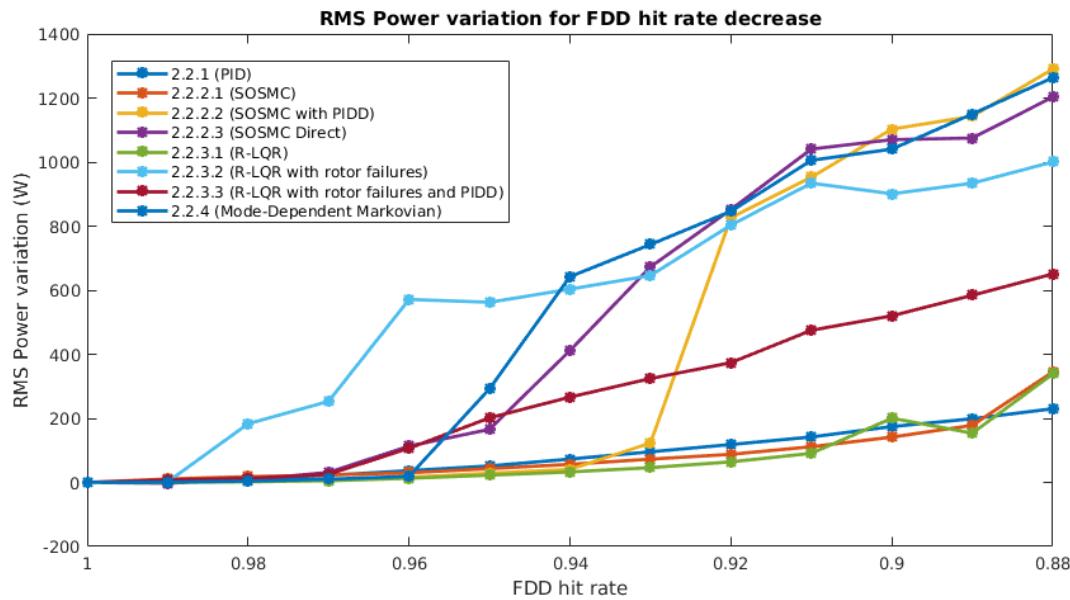


Figure 56: RMS Power variation to FDD delay increase.

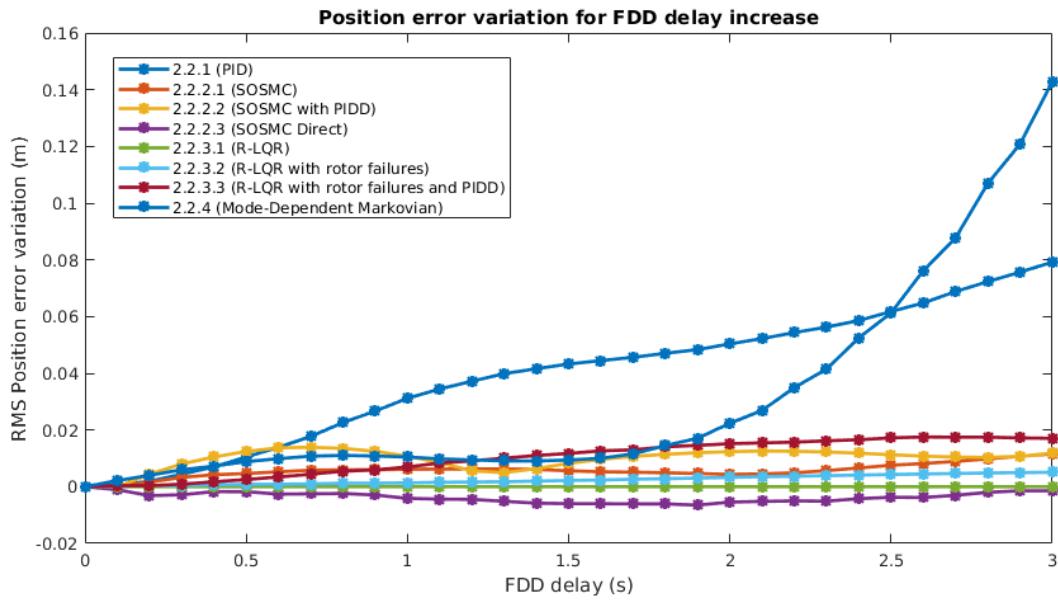


Figure 57: Position error variation to FDD hit rate decrease.

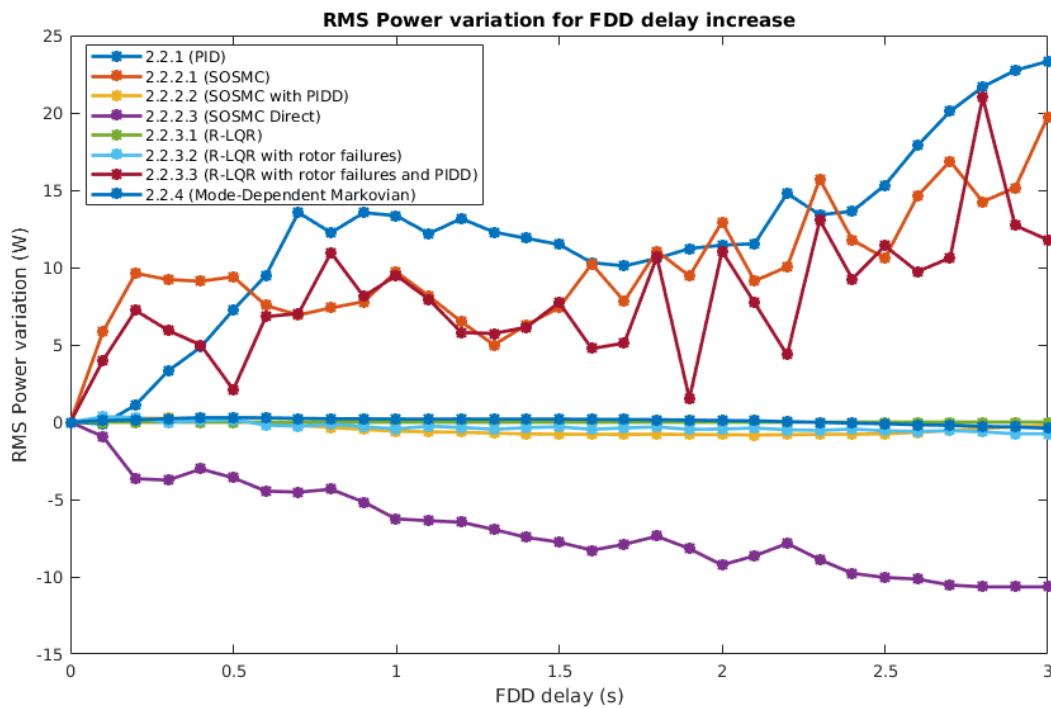


Figure 58: RMS Power variation to FDD delay increase.