# AN ALGORITHMIC FAULT-TOLERANT CONTROL ARCHITECTURE WITHOUT ACTUATOR REDUNDANCY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALP MARANGOZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
AEROSPACE ENGINEERING

SEPTEMBER 2018

Approval of the thesis:

# AN ALGORITHMIC FAULT-TOLERANT CONTROL ARCHITECTURE WITHOUT ACTUATOR REDUNDANCY

submitted by **ALP MARANGOZ** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Ozan Tekinalp
Head of Department, **Aerospace Engineering**

Assist. Prof. Dr. Ali Türker Kutay
Supervisor, **Aerospace Engineering Department, METU**

**Examining Committee Members:**

Assoc. Prof. Dr. İlkay Yavrucuk
Aerospace Engineering Department, METU

Assist. Prof. Dr. Ali Türker Kutay
Aerospace Engineering Department, METU

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Department, METU

Assist. Prof. Dr. Yıldıray Yıldız
Mechanical Engineering Department, Bilkent University

Assist. Prof. Dr. Kutluk Bilge Arıkan
Mechanical Engineering Department, TED University

**Date:**

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: ALP MARANGOZ

Signature :

# ABSTRACT

## AN ALGORITHMIC FAULT-TOLERANT CONTROL ARCHITECTURE WITHOUT ACTUATOR REDUNDANCY

Marangoz, Alp
Ph.D., Department of Aerospace Engineering
Supervisor : Assist. Prof. Dr. Ali Türker Kutay

September 2018, 160 pages

In this thesis work, a novel algorithmic fault tolerant control system architecture against actuator failures is developed. The method is based on injection of perturbations on the controlled states that are connected to healthy actuators, in order to compensate for the failed components and maintain overall stabilization of the system. An adaptive state estimator structure is used for detection of faults and fault mitigation perturbations are generated from a singularly perturbed dynamic system, which is a part of the control architecture. The proposed method is an algorithmic fault tolerant control architecture in a sense that the fault tolerance and reliability is achieved algorithmically and without using any redundant physical components.

For the theoretical analysis of the developed control system, the problem is formulated as a nonlinear control problem for interconnected systems and a theorem is structured that includes the assumptions, conditions and stability properties of the proposed architecture. Resultant algorithm can be applied to wide variety of problems including multi-input-multi-output unstable nonlinear systems, provided that the system under consideration is Lipschitz continuous and certain bound conditions are satisfied.

Design methodology is explained through theoretical analyses and analytically tractable numerical examples. Applications on more complex systems and limitations of the proposed fault tolerant control system architecture are demonstrated on joint failures of robotic manipulators and propeller loss of quadrotors cases through theoretical analyses and simulation results.

Keywords: Fault tolerant control, Nonlinear control, Adaptive control, Control of interconnected systems, Singular perturbation theory

# ÖZ

## YEDEK TAHRİK UNSURU KULLANILMAYAN BİR ALGORİTMA TABANLI HATA TOLERANSLI KONTROL MİMARİSİ

Marangoz, Alp
Doktora, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi : Dr. Öğr. Üyesi Ali Türker Kutay

Eylül 2018 , 160 sayfa

Bu tez çalışması kapsamında, tahrik elemanlarında gerçekleşen hataların kotarılmasına yönelik yeni bir algoritma tabanlı hata toleranslı kontrol mimarisi geliştirilmiştir. Geliştirilen metod, sağlıklı tahrik elemananlarına bağlı durumlara, hatalı elemanların hareketlerini telafi etmek ve sistemin genel kararlılığının sürdürülebilmesi amacıyla, pretürbasyonlar eklenmesine dayanmaktadır. Hata tespiti uyarlamalı durum kestiriciler ile gerçekleştirilirken hata kotarma amaçlı oluşturulan pertürbasyonlar ise kontrol sisteminin bir parçası olan tekil bir dinamik sistem tarafından üretilmektedir. Önerilen yöntemde hata kotarma amacıyla herhangi bir fiziksel yedekleme ihtiyacı duyulmaması nedeniyle, geliştirilen hata toleranslı kontrol mimarisi "algoritma tabanlı" olarak nitelendirilmiştir.

Geliştirilen mimarinin analizi için kontrol mimarisi, birbirine bağlı doğrusal olmayan sistemler için bir kontrol problemi olarak ele alınmış ve önerilen kontrol mimarisi ile ilgili varsayımları, koşulları ve kararlılık özelliklerini içeren bir teorem oluşturulmuştur. Oluşturulan yöntem, uygulanan sistemin Lipschitz sürekli olması ve bazı sınırları sağlaması koşulu ile, doğrusal olmayan, çok girdi-çıktılı ve kararsız sistemler de dahil olmak üzere birçok probleme uygulanabilmektedir.

Önerilen mimari için tasarım yöntemi, teorik analizler ve analitik olarak izelenebilecek sayısal örneklerle açıklanmıştır. Daha karmaşık sistemlerdeki uygulamaları ve yöntemin kısıtlamaları ise robotik manipülatörlerdeki eklem hatalarından kurtulma ve dört pervaneli hava araçlarındaki pervane kayıplarından kurtulma problemlerinde, teorik analizler ve sistem benzetim sonuçları ile beraber sunulmuştur.

Anahtar Kelimeler: Hata toleranslı kontrol, Doğrusal olmayan sistemlerin kontrolü, Uyarlamalı kontrol, Birbirine bağlı sistemlerin kontrolü, Tekil pertürbasyon teorisi

*To*

*Yıldız, Kanat and Eliz*

# ACKNOWLEDGMENTS

I would like to thank my supervisor Dr. Ali Türker Kutay for his support, understanding and guidance during my study. I also would like to thank thesis examining committee members Dr. İlkay Yavrucuk, Prof. Dr. Kemal Leblebicioğlu, Dr. Kutluk Bilge Arıkan and Dr. Yıldıray Yıldız for not only their time and effort for reviewing my work but also for their critical comments and key suggestions that I benefited for the final revision of this thesis.

My PhD study was a long journey and I had spent the first half of it under the supervision of Prof. Dr. Sinan Akmandor. I consider myself lucky for I had benefitted from his mentorship.

We shared a lot of this experience with my dear friend Dr. Mehmet Karaca. I am thankful to him and also to my friends Özer and Suzan Güvenç, Onur and Elif Ejdar Özel, Başak and Ahmet Kösereisoğlu for their continuous encouragement and belief throughout my study.

I am grateful to my dear family Çiğdem Marangoz, Suna Avcıl, Fatma Sumak, Mevlüt Sumak and Melis Sumak Hazır for their help, understanding, trust and encouragement.

And in the end, I owe it all to my little happy family for they give meaning and purpose to everything I do in my life. Without realizing, they make everything possible.

# TABLE OF CONTENTS

ABSTRACT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . v

ÖZ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . vii

ACKNOWLEDGMENTS . . . . . . . . . . . . . . . . . . . . . . . . . . x

TABLE OF CONTENTS . . . . . . . . . . . . . . . . . . . . . . . . . . xi

LIST OF TABLES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . xvi

LIST OF FIGURES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . xvii

LIST OF ABBREVIATIONS . . . . . . . . . . . . . . . . . . . . . . . . xx

CHAPTERS

1 INTRODUCTION . . . . . . . . . . . . . . . . . . . . . . . . . . 1

1.1 Motivation and problem definition . . . . . . . . . . . . . . 1

1.2 Control of interconnected systems . . . . . . . . . . . . . . . 3

1.3 Fault tolerant control systems . . . . . . . . . . . . . . . . . 6

1.4 Related control system design techniques . . . . . . . . . . . 9

1.4.1 Control system design methods based on compensation . . . . . . . . . . . . . . . . . . . . . 10

1.4.2 Singular perturbation theory based control methods 13

1.5 Proposed method, scope and limitations . . . . . . . . . . . 17

1.6 Summary of the thesis . . . . . . . . . . . . . . . . . . . . . 21

1.7 a Note on notation . . . . . . . . . . . . . . . . . . . . . 22

2 THEORETICAL DEVELOPMENT OF THE FAULT TOLERANT CONTROL STRUCTURE . . . . . . . . . . . . . . . . . . . . . . . . 25

2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . 25

2.2 Fault Mitigation as a Control Problem . . . . . . . . . . . . 28

2.3 Mathematical Preliminaries . . . . . . . . . . . . . . . . . . 30

2.3.1 Tikhonov's Theorem . . . . . . . . . . . . . . . . 30

2.3.2 Definitions related to stability of time-varying systems . . . . . . . . . . . . . . . . . . . . . . . . 31

2.3.3 Stability of time-varying and perturbed systems . . 33

2.3.4 Existence of solutions . . . . . . . . . . . . . . . . 35

2.4 Proposed Control Architecture . . . . . . . . . . . . . . . . 36

2.5 Numerical examples . . . . . . . . . . . . . . . . . . . . . 44

2.5.1 Example 1 . . . . . . . . . . . . . . . . . . . . . . 44

2.5.2 Example 2 . . . . . . . . . . . . . . . . . . . . . . 48

2.6 Adaptive state estimator as a fault diagnosis algorithm . . . . 51

2.7 Overview of the proposed method . . . . . . . . . . . . . . . 56

3 FAULT TOLERANT CONTROL OF ROBOTIC MANIPULATORS . 59

3.1 Problem definition . . . . . . . . . . . . . . . . . . . . . . 62

3.1.1 Governing Equations . . . . . . . . . . . . . . . . 62

3.1.2 Control of robotic manipulators . . . . . . . . . . 64

3.1.2.1 Computed torque technique . . . . . . 64

3.1.2.2 Partial-feedback linearization . . . . . 64

3.2 Fault diagnosis algorithm for robotic manipulators . . . . . . 66

3.2.1 Formulation . . . . . . . . . . . . . . . . . . . . . . 66

3.2.2 Selection of Basis Functions . . . . . . . . . . . . 69

3.2.3 Performance comparison . . . . . . . . . . . . . . . 70

3.2.3.1 Radial Basis Functions (RBF) Neural Network . . . . . . . . . . . . . . . . . 71

3.2.3.2 Chebyshev Polynomial based Orthogonal Set of Basis Functions . . . . . . 71

3.2.3.3 Custom Basis Set . . . . . . . . . . . 71

3.3 Design of the fault mitigation algorithm . . . . . . . . . . . 74

3.3.1 Solution of the perturbed system . . . . . . . . . . 76

3.3.2 Analysis of the boundary layer problem . . . . . . 77

3.4 Fault tolerant control architecture for robotic manipulators . . 78

3.5 Application on selected problems . . . . . . . . . . . . . . . 81

3.5.1 Vertical two-link robot arm . . . . . . . . . . . . . 81

3.5.1.1 Problem formulation . . . . . . . . . 83

3.5.1.2 Design of the fault detection algorithm 84

3.5.1.3 Design of the fault mitigation algorithm 85

3.5.1.4 Simulation results . . . . . . . . . . . 87

3.5.2 Horizontal three-link robot arm . . . . . . . . . . . 94

3.5.2.1 Problem formulation . . . . . . . . . 94

3.5.2.2 Design of the fault detection algorithm 96

3.5.2.3 Design of the fault mitigation algorithm 97

3.5.2.4 Simulation results . . . . . . . . . . . 97

3.6 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . 100

4 FAULT TOLERANT CONTROL OF QUADROTORS . . . . . . . . 103

4.1 Quadrotor control problem . . . . . . . . . . . . . . . . . . . 104

4.1.1 Equations of motion . . . . . . . . . . . . . . . . . . 106

4.1.2 Attitude stabilization . . . . . . . . . . . . . . . . . . 108

4.2 Reduced attitude dynamics . . . . . . . . . . . . . . . . . . . 110

4.3 Fault diagnosis algorithm for quadrotors . . . . . . . . . . . 114

4.4 Attitude Control with reduced number of propellers . . . . . 115

4.5 Fault tolerant controller for quadrotors . . . . . . . . . . . . 117

4.5.1 Perturbation relation . . . . . . . . . . . . . . . . . . 121

4.5.2 Stability of the boundary layer equation . . . . . . 122

4.5.3 Time dependent behaviour of the perturbation dynamics . . . . . . . . . . . . . . . . . . . . . . . . . 124

4.5.4 Overview of the design process . . . . . . . . . . . 125

4.6 Attitude stabilization examples with reduced number of propellers . . . . . . . . . . . . . . . . . . . . . . . . . . . . 127

4.6.1 Attitude Controller . . . . . . . . . . . . . . . . . . . 127

4.6.2 Fault Tolerant Controller Parameters . . . . . . . . 128

4.6.3 Simulation Results . . . . . . . . . . . . . . . . . . . 129

4.7 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . 132

5 CONCLUSION . . . . . . . . . . . . . . . . . . . . . . . . . . . 137

BIBLIOGRAPHY . . . . . . . . . . . . . . . . . . . . . . . . . . . . 141

APPENDICES

A MATRIX DIFFERENTIATION RULES . . . . . . . . . . . . . . . . 153

B LINEARIZATION OF ROBOTIC MANIPULATOR EQUATIONS . . 155

CURRICULUM VITAE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159

# LIST OF TABLES

TABLES

Table 3.1 Simulated two-link robot arm's parameters . . . . . . . . . . . . . . 87

Table 3.2 Simulated three-link robot arm's parameters . . . . . . . . . . . . . 98

Table 4.1 Fault diagnosis from the disturbance vector . . . . . . . . . . . . . 115

Table 4.2 Simulated system parameters . . . . . . . . . . . . . . . . . . . . . . 128

# LIST OF FIGURES

FIGURES

Figure 1.1 Factors affecting the nominal operation of a control system [25] . . 6

Figure 1.2 Typical $\mathcal{L}_1$ adaptive control architecture [38] . . . . . . . . . . . . 11

Figure 1.3 Typical Adaptive Model Inversion (AMI) - Concurrent Learning adaptive control architecture [41] . . . . . . . . . . . . . . . . . . . . . . . . 12

Figure 1.4 Typical Command Governor adaptive control architecture [46] . . . 13

Figure 1.5 Adaptive Dynamic Inversion (ADI) based control system architecture 16

Figure 1.6 Architecture of the developed fault tolerant control system . . . . . 18

Figure 2.1 Block diagram of the overall control system . . . . . . . . . . . . . 29

Figure 2.2 Schematic view of the bound on the states $x_1$. . . . . . . . . . . . . 42

Figure 2.3 Phase space trajectories of the example problem 1 . . . . . . . . . 46

Figure 2.4 Stabilization around the origin for the Example Problem 1 . . . . . 47

Figure 2.5 Trajectory tracking performance for the Example Problem 1 - a . . 47

Figure 2.6 Trajectory tracking performance for the Example Problem 1 - b . . 48

Figure 2.7 Stabilization around the origin for the Example Problem 2, with a diagonal reference model . . . . . . . . . . . . . . . . . . . . . . . . 52

Figure 2.8 Stabilization around the origin for the Example Problem 2, with a non-diagonal reference model . . . . . . . . . . . . . . . . . . . . . . . 52

Figure 2.9 Structure of the developed algorithmic fault tolerant control system 58

Figure 3.1 General schematic view of an open-chain robotic manipulator . . . 60

Figure 3.2 Comparison of basis function sets. . . . . . . . . . . . . . . . . . . . 73

Figure 3.3 Comparison of basis function sets with different parameters. . . . . 74

Figure 3.4 Fault tolerant control architecture for robotic manipulators . . . . . 79

Figure 3.5 Vertical two-link robot manipulator system . . . . . . . . . . . . . 82

Figure 3.6 Contour plot of $||P||_F$ for different values of $a$ and $b$. . . . . . . . . 88

Figure 3.7 Link position for the two-link robot arm problem simulation . . . . 90

Figure 3.8 Link angular velocities for the two-link robot arm problem simulation 90

Figure 3.9 Perturbation on the second links position ($r$), for the compensation of fault on the first joint . . . . . . . . . . . . . . . . . . . . . . . . . . 91

Figure 3.10 Applied torques for the two-link robot arm problem simulation . . 91

Figure 3.11 Moments on the links of the vertical two-link robot arm for the upward position . . . . . . . . . . . . . . . . . . . . . . . . . . . 92

Figure 3.12 Link position for the two-link robot arm problem simulation . . . . 94

Figure 3.13 Vertical two-link robot manipulator system . . . . . . . . . . . . . 95

Figure 3.14 Link positions after the occurrence of a fault for the three-link robot arm problem simulation without any fault mitigation act . . . . . . . . . . 98

Figure 3.15 Link position for the three-link robot arm problem simulation . . . 101

Figure 3.16 Perturbation on the second links position ($r$), for the compensation of fault on the third joint . . . . . . . . . . . . . . . . . . . . . . . . . . 101

Figure 3.17 Applied torques for the three-link robot arm problem simulation . . 102

Figure 4.1 Elements of a typical quadrotor control system . . . . . . . . . . . 105

Figure 4.2 Axis and direction definitions for the quadrotor problem. . . . . . . 107

Figure 4.3 Reduced attitude and related parameters. . . . . . . . . . . . . . . . 111

Figure 4.4 Quadrotor flight control system architecture, including the FTC algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 118

Figure 4.5 Maximum eigenvalue of $A_r$ for different yaw rates. . . . . . . . . . 124

Figure 4.6 Fault tolerant control architecture for quadrotors . . . . . . . . . . 127

Figure 4.7 Calculated disturbance vector for the quadrotor problem . . . . . . 130

Figure 4.8 Elements of the disturbance vector during the occurrence of the fault 130

Figure 4.9 Applied forces on the propellers, during the occurrence of the fault 131

Figure 4.10 Angle between the positive $z$ axis of the propeller and the local level plane . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 131

Figure 4.11 Yaw angle of the attitude vector on the $x - y$ local level plane . . . 132

Figure 4.12 Pitch rate perturbations . . . . . . . . . . . . . . . . . . . . . . . . 133

Figure 4.13 Euler angles of the quadrotor for different applied $f_{2,ftc}$ values . . . 134

Figure 4.14 Angular rates of the quadrotor for different applied $f_{2,ftc}$ values . . 135

Figure 4.15 Applied propeller thrusts for different $f_{2,ftc}$ values . . . . . . . . . 136

# LIST OF ABBREVIATIONS

| | |
|---|---|
| FMECA | Failure Modes, Effects and Criticality Analysis |
| FTC(S) | Fault Tolerant Controller (System) |
| LTV(S) | Linear Time Varying (System) |
| MIMO | Multi-input Multi-Output |
| RBF | Radial Basis Functions |
| SISO | Single-input Single-Output |
| SMC | Sliding Mode Control |

# CHAPTER 1

# INTRODUCTION

## 1.1 Motivation and problem definition

Any practical implementation of a control system requires some form of fault tolerance. Even robustness of control algorithms to variations in system parameters can be regarded as a fault tolerance measure, while for complex and safety critical systems a more systematic approach is required. Especially safety critical systems such as manned air vehicles and space launch systems can contain complex control architectures with various hardware and software redundancies in order to achieve certain level of reliability. But generally, there are two steps to be taken in order to make a system fault-tolerant: The existence of faults has to be detected and identified and the controller has to be adapted to the faulty situation so that the overall system continues to satisfy its goal [1]. This can be achieved through presence of redundant sensors and/or actuators. This redundancy can either be physical, *i.e.*, using more than one sensor and/or actuator per channel, or analytical. Analytical redundancies occur in the form of relations (or constraints) between states that are imposed by the system dynamics [2]. For example, jet engine under the wing of a transport aircraft both produces thrust and also pitching moment. Therefore limited control can be achieved in pitch channel with adjusting the thrust of an aircraft. Obviously, such kind of analytical redundancies are highly system specific.

Sensor costs are decreasing [3] and therefore for many systems, it is possible to use redundant sensors and increase the reliability in a cost-effective manner. However actuator redundancy is still costly, since actuators are in general more expensive and their presence have direct impact on physical design of systems. Therefore, a fault tolerant control system architecture, which does not require any actuator redundancy is very valuable. Furthermore, interest in low cost, small sized autonomous robotic systems are increasing. They are now used in more critical roles in operation with more complex and expensive payloads. Increasing reliability of these systems through usage of FTC architectures, without using redundant components is a critical part of development of low cost, safety critical autonomous systems.

The aim of this thesis work is development of such kind of an FTC architecture, *i.e.*, an FTC algorithm that exploits the dynamics of the system in order to achieve fault tolerance and reliability algorithmically, without using any redundant physical components.

The main idea for the developed FTC algorithm is to generate perturbations on the trajectories of the states that are connected to the healthy actuators that would compensate for the loss of the faulty actuators and stabilize the overall dynamics of the system. Proposed architecture is applicable to many system forms, including systems with unstable internal dynamics and fault detection is an integral part of the control system through adaptive components.

The idea of stabilizing perturbations can be formulated by the following Multi-Input-Multi-Output (MIMO) system:

$$\dot{x} = f(x, u) \quad \text{with,} \quad x, u \in \mathbb{R}^n$$

Consider the case where $m$ number of actuators are faulty. Then the states of the system can be grouped into two as $x_1 \in \mathbb{R}^m$ and $x_2 \in \mathbb{R}^{(n-m)}$ such that states that are included in $x_1$ are linked to the faulty actuators and the $x_2$ states are linked to the healthy actuators. With zero action on the faulty actuators, overall system dynamics can be cast into a cascaded form,

$$\dot{x}_1 = f_1(x_1, x_2) \tag{1.1a}$$

$$\dot{x}_2 = f_2(x_1, x_2, u_2) \tag{1.1b}$$

The proposed method can be regarded as introduction of a perturbation on the trajectory of $x_2$, so that the $x_1$ dynamics is also stable and stays in a bounded vicinity of origin.

In this thesis, a systematic methodology is developed for design of such a control system and a new fault tolerant control system architecture is proposed.

## 1.2 Control of interconnected systems

The dynamic system (1.1) covers a wide class of problems and it is often denoted as a “Cascade”, “Interconnected” or “Underactuated” system. It falls under the interest of sub-fields of control theory. For example, theoretical studies for the robust control of systems driven by external noise is based on the system representation where $f_1(x_1, x_2)$ is independent of $x_2$, *i.e.*, $f_1(x_1, x_2) = f_1(x_1)$. Similarly, adaptive control structures usually increases the order of the system with an additional dynamics for the calculation of the adaptive control inputs. Overall system driven by an adaptive controller can also be written in this form , where $x_1$ represents the adaptive controller's output and $u_2$ as the nominal controller's input.

The difficulty in finding general solutions for such systems comes from the result that even if both $\dot{x}_1 = f_1(x_1, 0)$ and $\dot{x}_2 = f_2(0, x_2, u)$ are globally asymptotically stable, the resulted interconnected system (1.1) might not be globally asymptotically stable. One way to guarantee overall stability is if both systems (1.1a) and (1.1b) are also input-to-state stable [4] ($x_2$ is regarded as an input for $f_1$ in this manner).

The overall system dynamics is closely related to the stability of the first system (1.1a), when input is zero, *i.e.*, stability properties of the system $\dot{x}_1 = f_1(x_1, 0)$. This is referred to as the “The zero dynamics” of the system (1.1). Systems whose zero dynamics is stable is called “minimum phase system” and it is called “non-minimum phase” if it is unstable.

It is known that output regulation with smooth feedback laws is possible for weakly minimum phase systems, provided that system has a relative degree one [1] [5]. Existence of a controller for the “minimum phase systems” with affine-in-control controlled system (1.1b) are studied by Marconi, Praly and Isidori. They have shown that it is possible to find a smooth controller that would solve the output regulation problem,

---

[1] Relative degree is defined as the number of times that the Lie derivative of the output function can be taken before the control input appears. For output function of identity, relative degree is always one.

provided that a feedback law exists that would keep the zero dynamics of the problem bounded and they have provided a method for construction of such kind of a controller in their subsequent work [6, 7].

For the "non-minimum phase systems", the unstability of the zero dynamics poses a problem and general methods for control system design is limited in the literature: Nazrulla and Khalil provided a method for systems in normal form. They used a sliding mode control on the estimate of the internal dynamics in order to achieve stabilization [8]. Chiang and Isidori on the other hand developed a regulator design technique which relies on a "post-processing" internal model and high-gain based stabilizers for single input multiple output systems [9]. Affine-in-control property, *i.e.*, when $f_2(x_1, x_2, u) = f_2'(x_1, x_2) \cdot u$, is important for the above mentioned results.

While above mentioned research concentrates on finding analytically exact methods for general class of systems, more pragmatic approaches are also available. Well known backstepping techniques involve systematically finding controllers for each state [10, 11]. Similar ideas are also available in the literature such as up/down augmentation [12] or dynamic surface control [13]. However, these analyses require specific forms for the dynamics such as, pure feedback or strict feedback form [14].

Another well known technique is the sliding mode controllers, where a stable manifold for the internal dynamic $\dot{x} = f_1(x_1, x_2)$ is found and switching based discontinuous control laws are employed in order to stabilize the overall system over this manifold [15]. Although some methods exists in order to smooth the control action, resultant control actions are nevertheless involve high frequency chattering.

When considering the internal dynamics (1.1a), the $x_2$ states can be regarded as a control input to the $x_1$ states. But for a general coupled nonlinear system, $x_2$ is not linearly related to the $x_1$ dynamics. Finding smooth feedback laws for non-affine, non-minimum phase systems, where the control signal is not linear is still a challenge, even for the case where the number of inputs is equal to the number of states. They are not directly applicable to systems in the form of (1.1), but following results can be mentioned for completeness of the literature review. Narang-Siddarth and Valasek achieved stabilization of open-loop unstable non-affine-in-control system with constructing the control input from superposition of a feedback function that renders the nonlinear system passive and an output feedback law that is used for stabilization [16]. Ho and Hedrick provided a systematic method to construct output vectors that transform the non-minimum phase output to a minimum phase one through linearization of the error dynamics for single input single output (SISO) non-minimum phase systems [17]. Some other selected techniques found in the literaturecan be summarized as, rewriting the nonlinear system in the state dependent coefficient form and solving the state dependent Riccati equation in each time step in order to calculate control inputs [18], approximating the non-affine-in-control function using neural networks [19–21] or fuzzy functions [22] and using adaptive update law for stabilization of the system.

Nonlinear control is a rich and very active research area. However, as shown in previous paragraphs, a general methodology that can be applied to all classes of problems are not available. Therefore, although proposed fault mitigation method can be cast into a control problem for interconnected systems, a novel control method that can be applied to wide range of problems is very valuable in this line of research.

Development of such a method, that can be applied to wide range of problems provided that certain conditions are satisfied, constitutes the thesis work. The perturbation signals are generated by a singular dynamic system in the proposed architecture (Figure 1.6). With this control structure, overall closed loop dynamics can be formulated as a singular perturbation problem and results of the Tikhonov's theorem can be used to analyze the behaviour of the system. Highlights of this FTC architecture -including assumptions, limitations and advantages-, scope of the thesis work and original contributions are summarized in Section 1.5, while formal and detailed analysis is provided in Chapter 2, together with a main theorem that covers the assumptions, limitations and results of the proposed method.

In order to compare the proposed method with the existing control structures in the literature, similar adaptive control architectures and control algorithms that rely on singular perturbation techniques is reviewed in Section 1.4, But before these discussions, a brief overview of fault tolerant control systems is given in next section, in order to introduce key concepts and nomenclature.

## 1.3 Fault tolerant control systems

Faults can occur within an operational life time of any system. Some faults critically damages the functionality of the system and therefore halts the operation of it, while others might just degrade some of the functionality and the system continues to operate with a reduced performance. Extends of the systems capability to continue operation under the affect of a fault is called Fault Tolerance [1].

Developments in fault tolerance control system analyses dates back to start of early implementation of control systems and it is also called "Execution Monitoring" in process control literature [23]. It is a research field on its own and there are many books [24, 25] and survey papers on the subject [26–29]. The aim in this section is to highlight briefly the key concepts of the problem. For this purpose, factors affecting the nominal operation of a control system is shown in Figure 1.1.

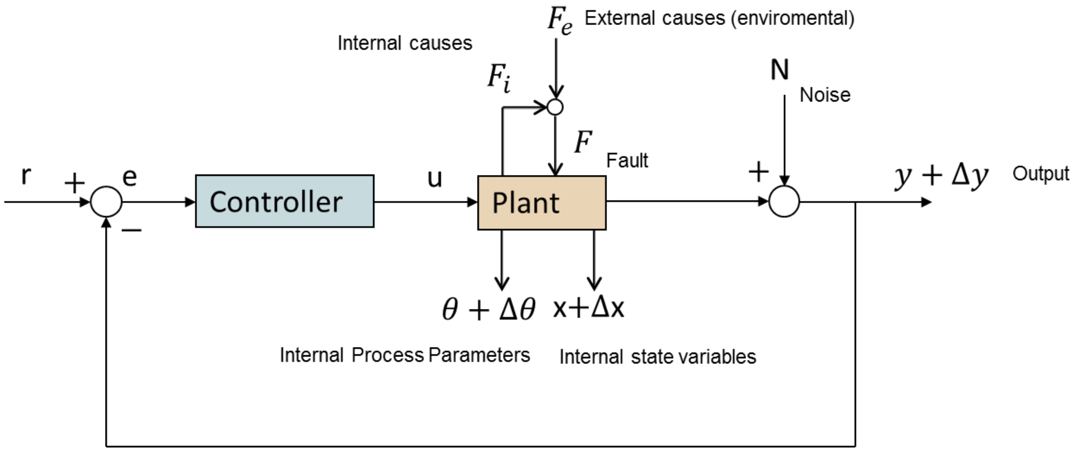


Figure 1.1: Factors affecting the nominal operation of a control system [25]

A control system is designed based on a modeled plant behaviour with certain parameter sets. The aim of the controller is regulate the outputs of the system, while controlling internal states of the plant. Of course, parameters and behaviour of the plant can deviate from the design conditions. Robustness analyses are used for maintaining the system states and outputs within operational limits, even under the presence of noise.

Such kind of deviations are rather continuous and do not introduce additional dynamics to the system. On the other hand, much more severe cases can be encountered in the form of faults. Control systems which are capable of tolerating potential faults in order to improve the reliability and availability while providing a desirable performance are called Fault Tolerant Control Systems (FTCS) [30].

Fault tolerant control techniques are usually system specific. In fact, one of the first steps of systems design includes a Failure Modes, Effects and Criticality Analysis (FMECA). In this part of the system design process, subsystems and system as a whole are investigated for failure possibilities and consequences of these failures on the operation of the system and safety of individuals. Design decisions are made based on these findings [1].

Results of such analyses are also used for development of fault tolerant control algorithms, which monitors the system operations in real time, determines if the system is functioning normally and executes predetermined actions if it is not, before failure becomes catastrophic. These functions are called “Fault Detection”, “Fault Diagnosis” and “Fault Mitigation”. These terms are explained below [1, 24, 25].

**Fault Detection** is the capability of the system for discriminating the presence of a fault from deviations from nominal operating conditions. Simplest form of fault detection algorithms is using output limits that causes system to stop when a certain threshold is crossed.

**Fault Diagnosis** is related to determination of the size and type or nature of the fault (Fault Identification), as well as the faulty component (Fault Isolation). This requires more information to be deduced from measurements and correct identification of the fault is critical for selecting the right response to the fault. For example in the case of robot arms, typical faults are locked joints or free swinging joints. With the knowledge of the specific faulty joint and the nature of the fault, it is possible to lock the motor brakes at the faulty joint (of course, if a lock mechanism is present) and find feasible trajectories for the end effector.

Effect of the fault on the system dynamics is dependent on the structure of the system and therefore fault diagnosis algorithms are highly system specific and usually requires a model for the fault, such as fault as an additive disturbance or a multiplicative parameter. Nevertheless, typical fault diagnosis algorithm structures that are related to the thesis work encountered in the literature are grouped and summarized below:

- State observers: Luenberger type state observers can be used for continuous estimation of the internal states of the system. Residuals of these observers bears information related to modeling error and external disturbances and therefore, they can be used for detection and identification of faults. Fault signals within the system can be constructed using basis functions, such as neural networks and this signal can be used for detection and identification of the fault.
- Nonlinear observers: System specific nonlinear observers can also be designed and used for fault estimation. While state observers require full state measurements for the estimation of the fault, nonlinear estimators can be designed specifically for reduced number of observations [31, 32].
- Multiple model extended Kalman filters / Filter banks: Kalman filters are extensively used in different applications for estimation purposes. For fault diagnosis, parallel running Kalman filter banks with different system models that correspond to specific fault model of the system, can be used. FTC algorithm can compare the outputs of these filters and find the best model that matches the observed outputs [2, 33].
- Stochastic models: Faults can be regarded as discrete changes in the state of the system. By assigning probabilistic measures, behaviour of the system can be modeled as Markov processes, with each state as a probable fault mode. Stochastic approach to the system dynamics is required for such fault diagnosis schemes [34].

**Fault Mitigation or Recovery** is the controllers actions for the system to continue its operation, with full or partial functionality. Fault mitigation is the key ingredient of an FTC scheme, since it is the whole purpose of designing an FTC structure. Typically two kinds of actions are done. If the fault changes the system parameters, parameters of the control algorithms are adjusted accordingly. Adaptive control systems inherently poses such kind of tuning capability but their operations might be unreliable [35]. Therefore, gain scheduling type controllers are also used. The other action in the event of a fault is reconfiguration of the system using healthy components. If it is a sensor fault, measurements of the faulty sensor are not used. Instead, operations are continued either using the measurements of a backup sensor, or the lacking information is completed using different sensor data. For example, pitch angle of an aircraft can be estimated using angle of attack sensors and GPS's velocity measurements, instead of a gyro. Such kind of relations are called analytical redundancies [2] and they can also be used for actuator failures. Although typical fault mitigation approach to actuator failures is to use redundant actuators and reconfigure the allocation of control signals to actuators [36] , many systems pose analytical redundancies in actuators. For example, position of the end point of a multiple link robot arm can be adjusted with more than one combination of the link angles. A feasible link angle combination can be found that both satisfies the constraints coming from the faulty link and the desired position of the end effector [37].

Fault tolerant control systems employ above mentioned functions on the control system in order to make the system robust to failures of components. Fault tolerant control is a mature field with different methodologies and many applications. Redundancy is widely used in employing of fault tolerant control systems but algorithmic fault tolerance measures that does not require physical redundancies is also sought, which is the central theme of this thesis work.

## 1.4 Related control system design techniques

The proposed method includes injection of high frequency perturbations on the trajectory of the actuated state and analyzing the stability of the control system using singular perturbation theory. Of course there are other techniques in the literature that uses superposition of control signals or singular perturbation theory as control method. Most of these developments are scattered in the literature and there is no single reference that overviews all of them. Therefore this section is dedicated to review of such related techniques.

### 1.4.1 Control system design methods based on compensation

A recurring idea in control system design techniques is generation of an auxiliary signal and superposing it to the output of a nominal controller. The auxiliary signal is usually used for online compensation of effects that had not been taken into account during the design of the nominal controller. Typical application of this idea is adaptive controllers but even Stability Augmentation Systems (SAS) of aircraft can be regarded in this way.

In fly-by-wire aircraft, flight control systems adds high frequency control signals generated by SAS to the low frequency commanding signals of the pilot, in order to maintain stability of the aircraft while tracking the commands of the pilot. Such kind of command superposition is possible for linear systems, since linear systems response to the superposition of two signals are itself superposition of the response of the system to the two input signals separately. Feedback signals also should be filtered for the frequency range of interest and fed to the relevant part of the controller, for the overall system to function properly.

For nonlinear systems, superposition of inputs in order to achieve an equivalent combined response is not possible. But many nonlinear control structures, especially adaptive controllers, use a similar structure to attain the control goal.

Consider the following MIMO dynamic system:

$$\dot{x} = A \cdot x + B \cdot (u + f(x))$$

where $f(x)$ represents the nonlinearity of the system.

In adaptive controllers, control signal is written as superposition of two signals: $u = u_n + u_{ad}$. The idea is to find control signal $u_{ad}$ that would cancel out the nonlinearity $f(x)$ so that resulting system becomes linear and the nominal control signal $u_n$ can be used for adjustment of this system. It is easier to prove that adaptive controllers converge to the desired negating signal $u_{ad} = -\hat{f}(x)$ asymptotically; but the real problem is, it is not always possible to do it with bounded control signals and with bounded error band. Bursting of control signals and temporal instabilities can be observed in many applications [35].

There exists novel adaptive control architectures that claims free of these problems. In $\mathcal{L}_1$ adaptive control architecture, Hovakimyan and Cao proposes to use projection based bounding functions on the estimation of the nonlinearity $(\hat{f}(x))$ and low pass filters to filter the high frequency part of the control system [38]. This makes $\mathcal{L}_1$ norms of the control signals bounded, which can be used in the design of the controller. Although some critics exists [39, 40], $\mathcal{L}_1$ adaptive control provides metrics that can be used for stability and transient performance prediction. Typical $\mathcal{L}_1$ adaptive control architecture is shown in Figure 1.2.

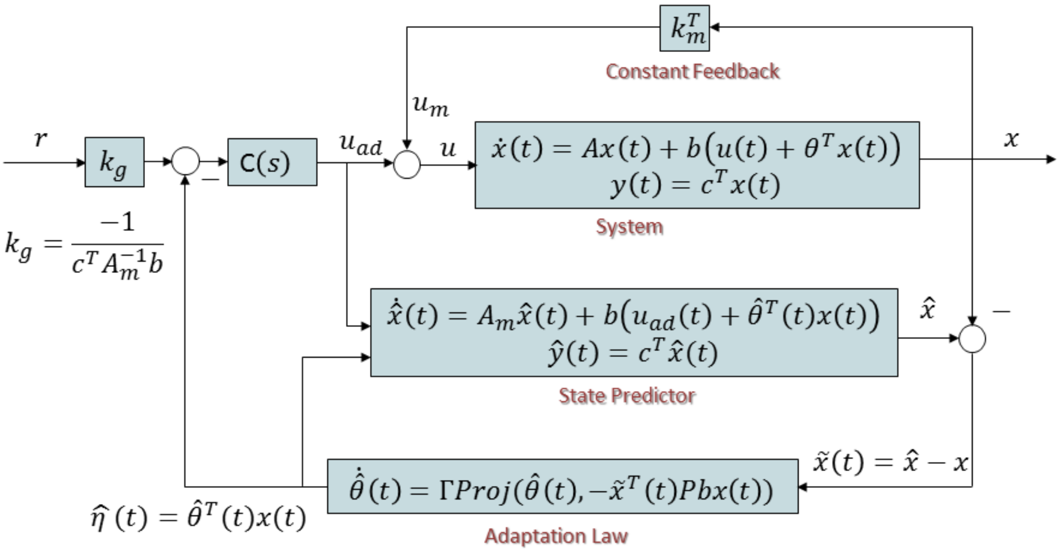

$k_m^T$
Constant Feedback
$u_m$
$r$
$k_g$
$-$
$C(s)$
$u_{ad}$
$u$
$\dot{x}(t) = Ax(t) + b\big(u(t) + \theta^T x(t)\big)$
$y(t) = c^T x(t)$
$x$
System
$k_g = \frac{-1}{c^T A_m^{-1} b}$
$\dot{\hat{x}}(t) = A_m \hat{x}(t) + b\big(u_{ad}(t) + \hat{\theta}^T(t)x(t)\big)$
$\hat{y}(t) = c^T \hat{x}(t)$
$\hat{x}$
$-$
State Predictor
$\tilde{x}(t) = \hat{x} - x$
$\dot{\hat{\theta}}(t) = \Gamma Proj(\hat{\theta}(t), -\tilde{x}^T(t)Pbx(t))$
$\hat{\eta}(t) = \hat{\theta}^T(t)x(t)$
Adaptation Law

Figure 1.2: Typical $\mathcal{L}_1$ adaptive control architecture [38]

Another approach is called Concurrent Learning adaptive control [41], whose implementation structure is shown in Figure 1.3. Estimation of nonlinearity $(\hat{f}(x))$ will be explained in Chapter 2, in detail; but it can be stated briefly that, this calculation is based on an error minimization algorithm that uses the instantaneous value of the error signal. In concurrent learning, historical stack of error signals are recorded and used for the estimation of the nonlinearity. Another idea for using recorded data is using orthogonal basis functions to approximate the nonlinearity and least square for the estimation of the coefficients of the basis functions [42].

It is known that performance of controllers can be improved by modifying the reference

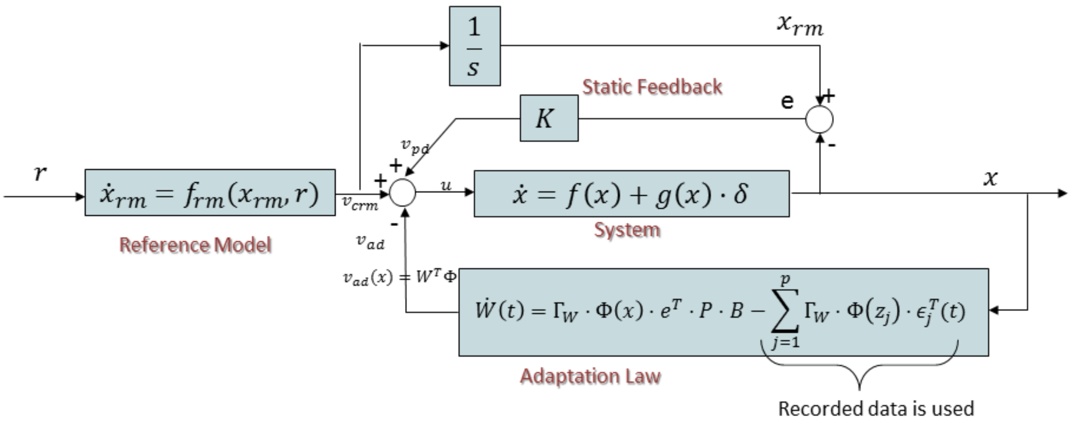

$\frac{1}{s}$
$x_{rm}$
Static Feedback
$K$
e
+
-
$v_{pd}$
+
+
$r$
$\dot{x}_{rm} = f_{rm}(x_{rm}, r)$
$v_{crm}$
$u$
$\dot{x} = f(x) + g(x) \cdot \delta$
$x$
Reference Model
-
$v_{ad}$
System
$v_{ad}(x) = W^T \Phi$
$\dot{W}(t) = \Gamma_W \cdot \Phi(x) \cdot e^T \cdot P \cdot B - \sum_{j=1}^{p} \Gamma_W \cdot \Phi(z_j) \cdot \epsilon_j^T(t)$
Adaptation Law
Recorded data is used

Figure 1.3: Typical Adaptive Model Inversion (AMI) - Concurrent Learning adaptive control architecture [41]

input of control systems, especially for enforcement of control constraints [43]. A regulator signal, so called the “Command Governor” signal is superposed on the reference input for this purpose. A command governor structure on an adaptive control architecture is also proposed by Yucelen and Johnson [44, 45]. The regulator signal is generated by an external linear system driven by the error between a reference model and actual output of the system. With this structure, controller becomes linear and therefore it is possible to define performance metrics together with signal bounds. Command governor modification is shown in Figure 1.4.

Adaptive control is not the only method that decomposes the control inputs and calculates them from different control objectives. It is known that high frequency oscillations can be used for trajectory tracking of underactuated robotic systems, if the system is in specific normal form and internal states are stable [47, 48]. Similarly, Schenato used high frequency oscillatory control signals superposed on the control input to control the flight of a robot insect [49]. Systems that don’t have internal dynamics, *i.e.*,$f(x) = 0$ in $\dot{x} = f(x) + g(x) \cdot u$, are called driftless systems. For such systems, averaging theory [4] provides relations between the response of the system to oscillatory inputs and the mean of the input signal. Using this relations, Schenato developed a feedback control design method which generates oscillatory control signals that can stabilize an unstable system, similar to the example provided in the introductory part.

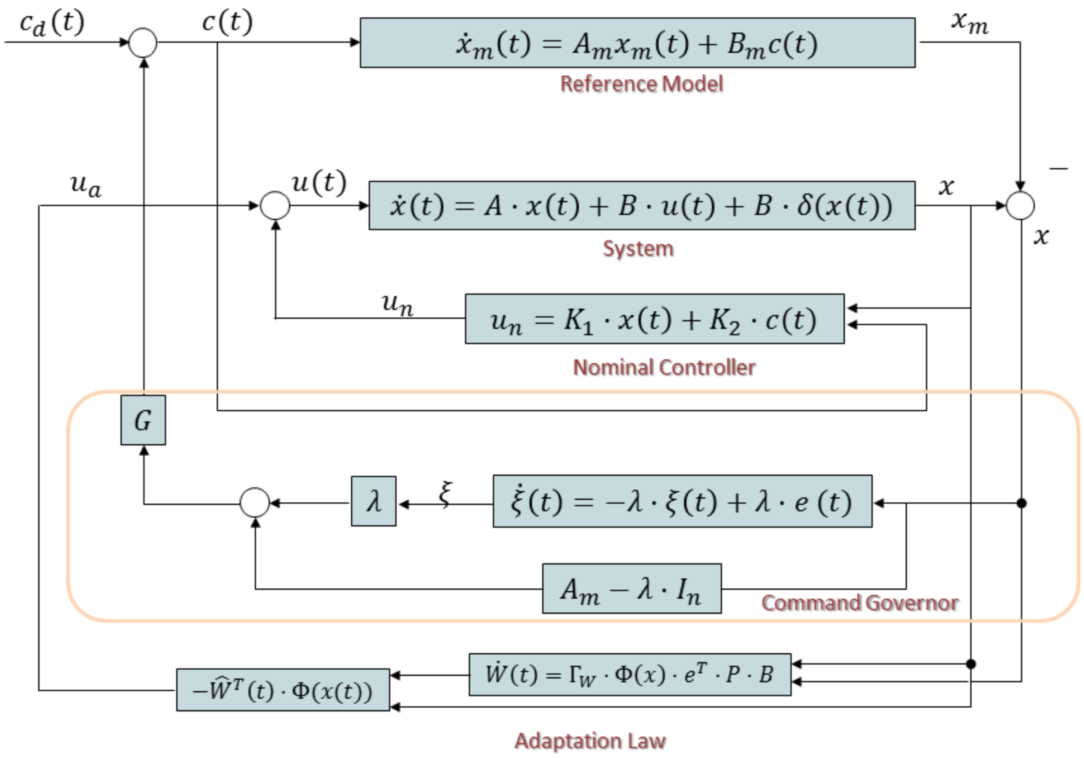

$c_d(t)$
$c(t)$
$\dot{x}_m(t) = A_m x_m(t) + B_m c(t)$
Reference Model
$x_m$
$u_a$
$u(t)$
$\dot{x}(t) = A \cdot x(t) + B \cdot u(t) + B \cdot \delta(x(t))$
$x$
$-$
System
$x$
$u_n$
$u_n = K_1 \cdot x(t) + K_2 \cdot c(t)$
Nominal Controller
$G$
$\lambda$
$\xi$
$\dot{\xi}(t) = -\lambda \cdot \xi(t) + \lambda \cdot e\,(t)$
$A_m - \lambda \cdot I_n$
Command Governor
$\dot{W}(t) = \Gamma_W \cdot \Phi(x) \cdot e^T \cdot P \cdot B$
$-\widehat{W}^T(t) \cdot \Phi(x(t))$
Adaptation Law

Figure 1.4: Typical Command Governor adaptive control architecture [46]

Control system design is a very rich and dynamic field and it is not possible to summarize, even briefly, all of the different techniques and applications within the content of a thesis. Those that are mentioned are the ones that have been found most relevant. Another aspect of the proposed FTC structure is usage of the singular perturbation theory, which will be reviewed in the next section.

### 1.4.2 Singular perturbation theory based control methods

Perturbation Theory deals with equations that have a small parameter -a perturbation- in their forms. Dependence of the solutions of algebraic, ordinary and partial differential equations on this small parameter can be analyzed using perturbation theory. In some cases, presence or absence of the perturbations changes the order of the equation, like if the coefficient of the highest derivative term in an ordinary differential equation is a small parameter. Such kind of problems are called singular perturbations and mathematical techniques that deal with such kind of systems are grouped as Singular Perturbation Theory.

Singular perturbation theory have many application areas. For control systems, these applications are mainly related to modeling of the plant that will be controlled. Using singular perturbation theories tools, one can simplify the problem and analyze the dynamics of the system at different time scales. Therefore singular perturbation theory can be used for include robustness analysis, time scale separation and system reduction [50].

Singular perturbation theory is used in this thesis work for analyzing the effects of high frequency perturbations on the system dynamics. Such a usage of singular perturbation theory, directly within the control law, is less common than the typical applications mentioned above. One of them is the Adaptive Dynamic Inversion (ADI) control system architecture developed by Hovakimyan, Lavretsky and Sasane [51, 52]. This technique relies on Tikhonov's theorem on singular perturbations [4]. It is one of the central theorems of the singular perturbations theory and also it is the starting point of the algorithmic fault mitigation approach developed in this thesis work. Therefore it will be discussed in Chapter 2 in detail, but general outline of the theory can be given by considering the following system of equations:

$$\begin{aligned} \dot{x} &= f(t, x, u, \epsilon) \quad (1.2) \\ \epsilon \cdot \dot{u} &= g(t, x, u, \epsilon) \quad (1.3) \end{aligned}$$

Now for $\epsilon \to 0$, (1.3) becomes $g(t, x, u, 0) = 0$. Let $u = h(t, x)$ be a solution to this equation. Then the dynamic system (1.2) and (1.3) can be reduced to

$$\begin{aligned} \dot{x} &= f(t, x, h(t, x), 0) \quad (1.4) \\ \frac{dy}{d\tau} &= g(t, x, y + h(t, x), 0) \quad (1.5) \end{aligned}$$

where $\tau = t/\epsilon$.

The first equation (1.4) is called the "Reduced System" and the second equation (1.5) is called the "Boundary Layer System". This reduction is valid provided that the system is comprised of continuous functions, reduced system has a unique solution and the origin of the boundary layer system is exponentially stable.

The idea of the ADI technique is that if the perturbed dynamics (1.3) is taken as the control input, then Tikhonov theorem provides a basis for reduced system (1.4) to become the inversion of the original system. Let $f_{rm}(t,x,r)$ be the reference model that we want our system to follow, with $r$ being the reference input. By choosing the right hand side of the perturbed dynamics as $g(t,x,u) = f(t,x,u) - f_{rm}(t,x,r)$, results of Tikhonov's theorem states that control input $u$ would lead the system into the state $x$ where $f(t,x,u) = f_{rm}(t,x,r)$, *i.e.*,practically control input would invert the system.

In order to satisfy the requirements on the stability of the reduced and boundary layer system, following form of equations is used .

$$\begin{aligned} \dot{x} &= f(t,x,u) \\ \epsilon \cdot \dot{u} &= -sign(\frac{\partial f}{\partial u}) \cdot [f(t,x,u) - (A_{rm} \cdot x_{rm} + B_{rm} \cdot r)] \end{aligned} \tag{1.6}$$

where reference model dynamics is $\dot{x}_{rm} = A_{rm} \cdot x_{rm} + B_{rm} \cdot r$ and $-sign(\frac{\partial f}{\partial u})$ ensures that the boundary layer system is stable. A critical assumption for the controller is that $\frac{\partial f}{\partial u}$ is sufficiently bounded away from zero for all $(t,e) \in [0,\infty)$.

Continuing work by Lavretsky and Hovakimyan extends the formulation to compensate for uncertainties through online estimation of the nonlinearity with a neural network structure [52]. For this purpose, a state estimator and adaptive update law is added to the control structure.

For the control of the system in the form of $\dot{x} = A \cdot x + B \cdot f(t,x,u)$, nonlinearity $f(t,x,u)$ can be represented by linear combination of basis functions. Radial basis function (RBF) based neural networks are typically used for such purposes. Let $\Phi(x,u)$ be $n$ dimensional basis function and $W(t)$ be the vector of $n$ weighting coefficients, then the approximation of $f(t,x,u)$ can be written as $\hat{f}(t,x,u) = \hat{W}(t)^T \cdot \Phi(\hat{x},u)$. This approximate form is used in the right hand side of (1.6). Adaptive update law used for updating of weighting coefficients is given in (1.7), where $\Gamma$ is the adaptation gain, $e_s = \hat{x} - x$ is the estimation error and $P$ solves the algebraic Lyapunov equation $A_s^T \cdot P + P \cdot A_s = -Q$ for arbitrary $Q = Q^T > 0$ and $A_s$ is the stable state estimation error dynamics. In order to bound the weighting coefficients, projection operator is applied on the adaptive update law. Together with the state estimator (1.8) and control law (1.9), final form of the control system can be achieved (Figure 1.5).

$$\dot{\hat{W}}(t) = \Gamma \cdot \text{Proj}\left(\hat{W}(t), -\Phi(\hat{x}, u) \cdot e_s \cdot P \cdot B\right) \tag{1.7}$$

$$\dot{\hat{x}}(t) = A \cdot x + B \cdot \left(\hat{W}(t)^T \cdot \Phi(\hat{x}, u) + A_s \cdot e_s\right) \tag{1.8}$$

$$\epsilon \cdot \dot{u} = -sign(\frac{\partial f}{\partial u}) \cdot \left[\hat{W}(t)^T \cdot \Phi(\hat{x}, u) - (A_{rm} \cdot x_{rm} + B_{rm} \cdot r) + A_s \cdot e_s\right] \tag{1.9}$$

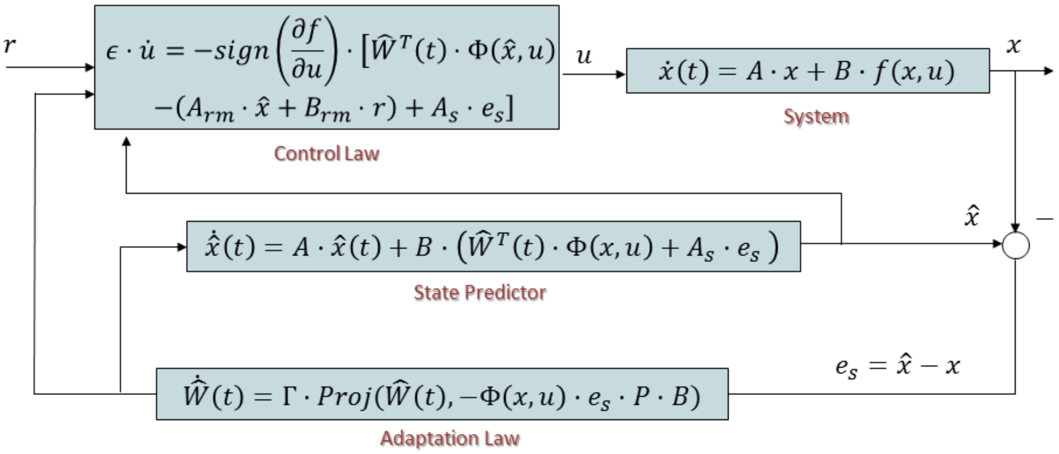

$r$
$\epsilon \cdot \dot{u} = -sign\left(\frac{\partial f}{\partial u}\right) \cdot [\widehat{W}^T(t) \cdot \Phi(\hat{x}, u)$
$-(A_{rm} \cdot \hat{x} + B_{rm} \cdot r) + A_s \cdot e_s]$
Control Law
$u$
$\dot{x}(t) = A \cdot x + B \cdot f(x,u)$
System
$x$
$\hat{x}$
$-$
$\dot{\hat{x}}(t) = A \cdot \hat{x}(t) + B \cdot (\widehat{W}^T(t) \cdot \Phi(x,u) + A_s \cdot e_s)$
State Predictor
$e_s = \hat{x} - x$
$\dot{\widehat{W}}(t) = \Gamma \cdot Proj(\widehat{W}(t), -\Phi(x,u) \cdot e_s \cdot P \cdot B)$
Adaptation Law

Figure 1.5: Adaptive Dynamic Inversion (ADI) based control system architecture

For a further improvement, integrals of the RBFs, together with RBFs as basis functions can be used which assures that assumption on monotonic property with respect to control input of the unknown system dynamics is satisfied [53].

However, further analysis of Teo and How revealed that ADI controller is equivalent to a linear proportional-integral model reference controller if the sign of the control effectiveness is known [54]. This can be seen (roughly) by considering the integral of the control law in (1.6) and using the relations $\dot{x} = f(t, x, u)$ and $\dot{x}_{rm} = A_{rm} \cdot x_{rm} + B_{rm} \cdot r$:

$$\begin{aligned}
&\int_0^t \epsilon \cdot \dot{u} \cdot dt = \int_0^t f(t, x, u) \cdot dt - \int_0^t (A_{rm} \cdot x_{rm} + B_{rm} \cdot r) \cdot dt \\
&\rightarrow \epsilon \cdot u(t) = \int_0^t \dot{x} \cdot dt - \int_0^t \dot{x}_{rm} \cdot dt \\
&\quad \rightarrow u(t) \approx \frac{1}{\epsilon} \cdot (x(t) - x_{rm})
\end{aligned}$$

Therefore, this form of ADI is equivalent to calculation of the difference between a reference model $x_{rm}$ and the measured state $x$ and multiplying with a large gain $1/\epsilon$.

However this equivalence holds only for the time response when applied to the exact system and robustness properties of ADI and PI realization are not equivalent. In fact proportional-integral realization performs better when accurate knowledge of the nonlinear system dynamics is not available [55], while ADI realization is more robust to time delays [56].

Although further research on ADI is not pursued, there exists few works in the literature that use a similar idea. For example, higher order derivatives of the control signal and the state feedback signal can be used for generation of the control signal [57]:

$$\epsilon^q \cdot u^{(q)} + d_{q-1} \cdot \epsilon^{q-1} \cdot u^{(q-1)} + \ldots + d_1 \cdot \epsilon \cdot \dot{u} + d_0 \cdot u = k_0 \cdot \left( f_{rm} - x^{(n)} \right)$$

where superscript $(n)$ represents the $n$-th time derivative.

In another application, control law is written as a singularly perturbed system not for the inversion of the whole system, but just for the inversion of the non-affine control function [58]. For the system in the form of $\dot{x} = f(x) + g(x, u)$, control law written in the form of $\epsilon \cdot \dot{u} = -\left( g(x, u) - v \right)$ would lead to the converged system of $\dot{x} = f(x) + v$.

Singular perturbation theory provides a method for generation of fast control signals and time scale separation for the control system analysis and this feature is exploited in the above mentioned methods. In this thesis work, a similar idea is constructed for the development of an algorithmic fault mitigation approach, but instead of generating control signals, singular perturbation theory is used for generation of reference trajectories.

### 1.5 Proposed method, scope and limitations

Recovery from actuator failures, without using redundant actuators is aimed for the thesis work. For this purpose, an algorithmic fault tolerant control architecture is developed. This architecture is basically an adaptive controller that generates perturbations on the trajectories of the states that are connected to healthy actuators, that would stabilize or guide the motion of the states that are connected to faulty actuators. Fault detection and isolation is achieved through state estimator as part of the algorithm. Overall system can be implemented within a Health Monitoring framework, *i.e.*,as an external outer control system that checks the operation of the control system and warns and/or executes predetermined functions if a fault is detected. General overview of the system is shown in Figure 1.6. Here, $u_1$ represents the faulty actuator and perturbations on the reference trajectory of the actuated states ($r_2$) are used for stabilization of the whole system. It should be remarked that states ($x_1, x_2$) and control signals ($u_1, u_2$) shown can be multidimensional, although they are grouped into two as those that are connected to healthy actuators ($x_2$) and those that are connected to faulty actuators ($x_1$).

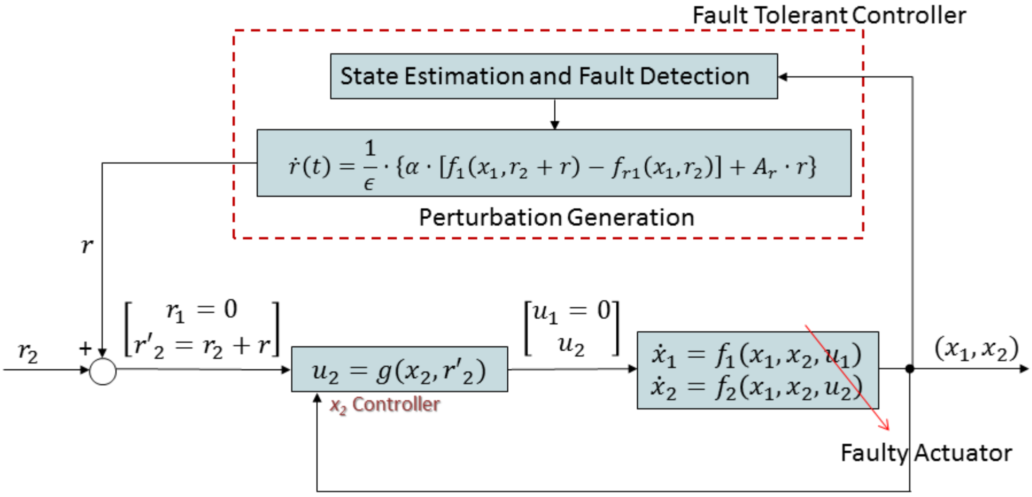

Fault Tolerant Controller
State Estimation and Fault Detection
$\dot{r}(t) = \frac{1}{\epsilon} \cdot \{\alpha \cdot [f_1(x_1, r_2 + r) - f_{r1}(x_1, r_2)] + A_r \cdot r\}$
Perturbation Generation
$r$
$r_2$
+
+
$\begin{bmatrix} r_1 = 0 \\ r'_2 = r_2 + r \end{bmatrix}$
$u_2 = g(x_2, r'_2)$
$x_2$ Controller
$\begin{bmatrix} u_1 = 0 \\ u_2 \end{bmatrix}$
$\dot{x}_1 = f_1(x_1, x_2, u_1)$
$\dot{x}_2 = f_2(x_1, x_2, u_2)$
$(x_1, x_2)$
Faulty Actuator

Figure 1.6: Architecture of the developed fault tolerant control system. Within the shown configuration, perturbations on the reference trajectory of the actuated states ($x_2$) are used for stabilization of the whole system

As the details will be explained in Chapter 2, the problem is formulated such that the stabilizing perturbations are generated from an external dynamic system and overall closed loop system can be written as a singular perturbation problem:

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2) \\ \dot{x}_2 &= f_2(x_1, x_2, u) \\ \epsilon \cdot \dot{r} &= \alpha \cdot [f_1(x_1, r + r_2) - f_{r1}(x_1, r_2)] + A_r r \end{aligned}$$

where $f_{r1}(x_1, x_2)$ is the reference model for the $x_1$ dynamics and $r_2$ is the reference signal for $x_2$. $A_r$ is a Hurwitz matrix , $\epsilon$ is a small parameter and $\alpha = \pm 1$ will be chosen appropriately for stable $r$ dynamics. Choice of $\epsilon$, $f_{r1}$ and $A_r$ are part of the fault tolerant control system design process.

The proposed control law can be regarded as introduction of an additional state as a secondary command on the trajectory of $x_2$. This secondary command is generated by a singular dynamic system, whose structure is determined in order to render the $x_1$ dynamics into a desired form. With this control structure, overall closed loop dynamics can be formulated as a singular perturbation problem and results of the Tikhonov's theorem (Theorem 2.3.1) can be used to analyze the behaviour of the system.

As will be shown in Chapter 2, proposed control structure can be applied to wide class of problems, including systems with unstable internal dynamics, *i.e.*, $\dot{x}_1 = f_1$ is unstable, provided that the systems under consideration satisfy the following assumptions:

**Assumptions**

- Proposed method is suitable for continuous systems, including the closed loop controller for $x_2$ dynamics. Therefore, discontinuous control laws, such as Sliding-Mode-Controllers or switching control are not allowed in the original system.
- The structure of the system should permit that the perturbations on the directly controlled states ($x_2$) affect the $x_1$ dynamics. This condition can be formulated with the following controllability assumptions:
  - The homogeneous system $\dot{x}_1 = f_1(x_1, x_2)$, with $x_2$ as the control input, is small-time locally controllable from $x_{1,0} = 0 \; \forall x_2 \in \mathbb{R}^m : x_2 \subset \Omega$.
  - The homogeneous system $\dot{x}_2 = f_2(x_1, x_2, u)$ is small-time locally controllable from $x_{2,0} = 0 \; \forall x_1 \in \mathbb{R}^{(n-m)} : x_1 \subset \Omega$.
- In order to prove that the perturbation signal $r$ is bounded, a certain condition on $\dot{x}_1 = f_1(x_1, x_2)$ dynamics is required, which can be formulated as an assumption on the existence of a the following bound:

$$\|f_1(0, x_2)\| \leq \gamma_2 \|x_2\|$$

Details of this condition is discussed in Chapter 2. Note that this condition does not restrict that $\dot{x}_1 = f_1(x_1, 0)$ should be stable. In fact the system can be unstable even with respect to $x_2$, but the growth should be linearly bounded.

- Initial conditions of the system should be bounded.

### Limitations

Proposed method is applicable to wide class of problems, provided that the above mentioned assumptions are satisfied. The limitations on the results that can be achieved can be summarized as follows.

- Proposed methodology guarantees the stability of the uncontrolled dynamics ($x_1$) such that it is ultimately bounded. It does not guarantee convergence and proposed methodology does not provide a mechanism for command tracking of the unactuated states.
- Although it is proved that the trajectory perturbations are bounded, provided that the above mentioned assumptions are valid, their bound, which also depends on initial conditions and value of $\epsilon$, might exceed actuator or system limits. But it should be noted that the results provided in Chapter 3 and 4 are achieved using system simulations that takes into account these effects as well.
- Transient response of the system have not been taken into account in the analyses, for its mathematical treatment in general nonlinear systems is difficult, it is left outside of the scope of the thesis work. Since the method is developed as a fault tolerance measure, for the case under study, stability is more important than performance.

### Advantages

With these assumptions and limitations, advantages of the proposed method can be summarized as follows.

- Proposed FTC architecture does not require any actuator redundancy.

- It can be used in systems where number of actuators are less then number of states, *i.e.*, underactuated systems.
- It can also used in systems which have unstable internal dynamics.
- It can be implemented as an outer loop to an existing control system and it can function without using switching laws.
- Fault detection is an integral part of the FTC system.

### Original Contributions

A novel algorithmic fault tolerant control architecture is developed within this thesis work. Original contributions can be summarized as follows.

- Developed FTC is in fact an adaptive controller and therefore it can be used directly as a control algorithm for control of nonaffine in control, cascade (or interconnected) systems that have unstable internal dynamics. Within the limitations explained in Chapter 2, this capability is unique to the proposed architecture.
- Due to coupling of system dynamics, estimation of disturbance signal is difficult in robotic systems. Within the content of the thesis work, a state transformation is proposed for robotic systems, which decouples and enables correct construction of the external disturbance signal.
- Fault tolerant controller for quadrotors that deal with propeller loss is very limited in the literature. Proposed solution is a valuable contribution that it can be implemented together with an existing controller and without requiring any linearization.

## 1.6 Summary of the thesis

Development of a new fault tolerance control architecture and laying out the design methodology constitutes the thesis work. In order to explain the details of the proposed method, the rest of the thesis is organized as follows:

**Chapter 2.** Theoretical foundations of the proposed algorithmic fault tolerant control architecture is laid down in Chapter 2. For this purpose, it is shown that how fault tolerant control problem can be cast into a control problem for nonlinear interconnected systems. Details of the proposed control architecture are explained with stability proofs. Applications on model problems are also provided.

**Chapter 3.** Robotic systems can be cast into a standard second order Lagrangian form, independent of the dimension of the system and most of the robotic control systems use this form as the model of the plant dynamics. FTC algorithm is derived for this general form and applied to a vertical two-link and a horizontal three-link robot arm problems in Chapter 3, in order to demonstrate the feasibility of proposed FTC architecture on robotic systems. A suitable transformation of states for fault detection in robotic systems is also presented.

**Chapter 4.** Attitude control is an essential part of any flight control system. Quadrotor is chosen as an application platform and developed FTC algorithm implemented for complete loss of a propeller case.

**Chapter 5.** Finally, remarks on the proposed algorithmic fault tolerant control architecture, further research directions and possible applications are given in Chapter 5.

## 1.7 a Note on notation

The focus of this thesis is Multi-Input-Multi-Output systems and therefore formulations are applicable to multidimensional systems. No additional symbol is used in order to indicate multidimensionality and unless otherwise specified, all variables are vector variables. As usual, multiplication of variables is implied with consequent placement, without any operator in between. In order to improve the readability of some formulations, "$\cdot$" sign is also used as indication of multiplication operation. For equations involving matrix terms, all multiplications are matrix multiplication.

For vector product (or cross product) between vectors, "$\times$" sign is used. In some formulations, "$\times$" operator is used in front of vectors to indicate the skew-symmetric form of vector. It is defined for $u \in \mathbb{R}^3$ as,

$$(u\times) = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}$$

# CHAPTER 2

# THEORETICAL DEVELOPMENT OF THE FAULT TOLERANT CONTROL STRUCTURE

Main output of this thesis work is a novel control structure that can be used as an algorithmic fault tolerance measure. Therefore a controlcentric theoretical analysis is required in order to determine the design parameters and range of applicability of the proposed structure.

For this purpose, first, how the proposed method can be formulated as a control problem for interconnected or cascade systems is explained in Section 2.2. Then in Section 2.3, some stability related definitions, theorems and results that will be used in the analysis is reviewed. After that, the main theorem that covers the extend, assumptions and results of the proposed method is given in Section 2.4.

Fault detection is an integral part of the proposed method and its structure is explained in Section 2.6. In order to demonstrate the performance of the control system, analytically tractable prototype problems are discussed in Section 2.5 and finally, general overview of the method is summarized in Section 2.7.

## 2.1 Introduction

Fault tolerant control problem can be analyzed using a general form of a nonlinear Multi-Input-Multi-Output (MIMO) dynamic system (2.1) with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^r$.

$$\begin{aligned} \dot{x} &= f(x) + g(x,u) \\ y &= h(x,u) \end{aligned} \tag{2.1}$$

Generally, three types of faults can be considered for a development of an FTCS:

- Many systems are designed for finite number of operation points and control parameters are tuned for the specified conditions. Presence of a fault might lead to sudden change of parameters or introduction of unmodeled dynamics such as appearance of vibration. This kind of faults are related to unknown changes in $f(x)$ and $g(x,u)$.
- Control systems use sensors in order to gather information about the states of the system. Faults in the sensor measurements are possible which would lead to faulty readings or no measurements at all. This can be analyzed through analysis of the measurement function $h(x)$ and output vector $y$.
- Actuators are the mechanism that control systems use in order to manipulate the plant and unlike sensors, whose dynamics can be safely ignored in most applications, they an integral part of the system. Actuator faults may lead to partial or complete loss of control capability and in terms of control synthesis, a major change in $g(x,u)$ dynamics occurs.

In order to evaluate the effects of failures and develop control algorithms for coping with them, different tools from the control system theory can be used.

Change in the dynamics of the system is rather a robust control problem. Various analysis and design techniques exist in the literature to make the control system cope with such variations and adaptive control is one of them. With implementing adaptive components in the FTC structure, proposed control system exhibits a fault tolerance character in this manner.

If $r > n$ in (2.1), then there are more measurements than required and this can be used for finding and replacing a faulty sensor. In typical applications, Kalman filters are employed to find an optimal estimate of the measurement and bounds on probable value of the signal or multiple measurements are voted for their validity. Such kind of algorithms can be implemented without effecting the rest of the control system.

Another approach is using physical constraints of the system. In many cases, physics of the problem pose restrictions on the states. For example, no-slip condition on a rolling wheel imposes a constraint between the angular velocity of the wheel and the speed of the vehicle it carries. These are called "Holonomic constraints" and they are very common in robotic systems. Such constraints can be used as rules for checking the validity of the sensor measurements. Since these constraints are system specific, resulted FTC algorithm against sensor faults is also system specific.

Since most of the fault tolerance measures against sensor faults are either too system specific or can be developed outside of the control system, such measures are not addressed in this thesis work. But the developed FTC architecture does not inhibit implementation of such measures.

Main aim of the developed FTC algorithm is to propose a systematic design methodology to deal with actuator faults with the reduced capability of the control system with wide applicability. The proposed solution to the problem is:

- Fault diagnosis through estimation of the time dependent fault signal

- Zeroing the control input of the faulty actuator as the first fault mitigation act.

- Generating perturbations on the healthy actuators for overall stabilization and command following of the damaged system.

In order to analyze the proposed implementation of the above fault mitigation algorithm, it will be explained in Section 2.2 that such kind of a fault mitigation algorithm would result in a coupled system where number of actuating signals are less than number of states (assuming no overactuation in the original system), which are also commonly called as cascade systems or interconnected systems. As explained in Section 1.2, existing control system architectures for such kind of systems have a limited range of applicability. Therefore a novel control system architecture is developed. This control system architecture is analyzed in subsequent sections.

## 2.2 Fault Mitigation as a Control Problem

In order to pose the algorithmic fault mitigation policy as a control problem, let the number of faulty actuators be $m_f$. For the case where number of states is equal to or more than the number of actuators ($n \geq m$), the states $x \in \mathbb{R}^n$ and control input $u \in \mathbb{R}^m$ of (2.1) can be split into two as $(x_1, x_2)$ and $(u_1, u_2)$ where $x_1 \in \mathbb{R}^{n-(m-m_f)}, x_2 \in \mathbb{R}^{m-m_f}, u_1 \in \mathbb{R}^{m_f}, u_2 \in \mathbb{R}^{m-m_f}$. The aim of this grouping is to separate the faulty actuators and write down the dynamics of the healthy actuators as a fully actuated system. However for a coupled nonlinear system, splitting the states is rather arbitrary. When the mechanics of the problem is such that an actuator is directly related to a state, as in the case of the robot arm problem, it is logical to choose $x_1$ that are directly linked to $u_1$ but in general, any $m_f$ number of states can be chosen as the internal dynamics. Since this choice may effect the performance of the controller, alternatives should be evaluated during the control system design process.

With this note, (2.1) can be rewritten as,

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, u_1 + d) \\ \dot{x}_2 &= f_2(x_1, x_2, u_2) \end{aligned} \tag{2.2}$$

The fault signal is modeled as an additive signal to $u_1$ and it is denoted as $d$. The nature of the fault determines the fault signal $d$. For example if the actuator becomes ineffective, then the fault signal can be written as $d(t) = -u_1(t)$. Or if there is an unexpected lag in the actuator, fault signal can be modeled as $d(t) = -u(t) + u(t+\tau)$, with $\tau$ denoting the time lag. In any case, fault signal is a function of control input, *i.e.*, $d = f(u_1(t))$. With this approach, fault diagnosis algorithm becomes estimation problem for the signal $d(t)$ and the fault mitigation problem becomes control under the effect of an external disturbance.

In order to cope with the faulty situation, first fault mitigation act is nullifying the effect of the fault. This can be done with cutting of the power of the actuator or locking the actuators. Either of the actions result in $u_1 = d = 0$. Although actuator lock in a nonzero position might result in a residual control force, which can be regarded as an uncertain time independent constant disturbance. Such effects can be handled through adaptive parts of the FTC algorithm.

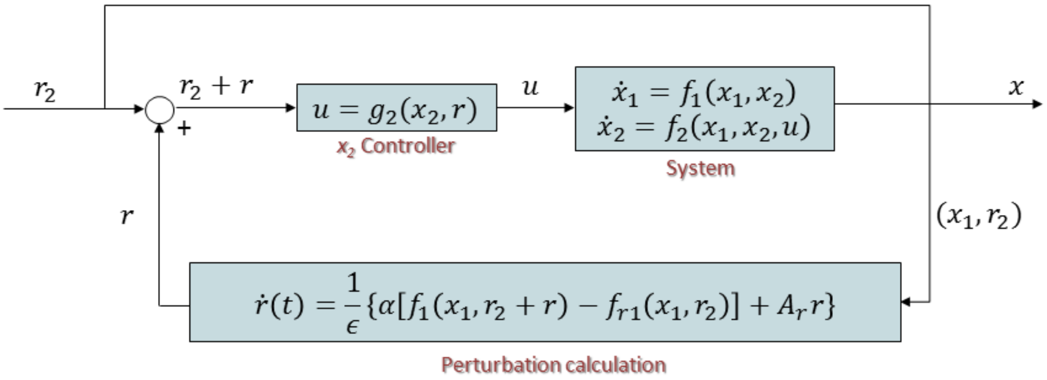

$r_2$
$r_2 + r$
+
$u = g_2(x_2, r)$
$x_2$ Controller
$u$
$\dot{x}_1 = f_1(x_1, x_2)$
$\dot{x}_2 = f_2(x_1, x_2, u)$
System
$x$
$r$
$(x_1, r_2)$
$\dot{r}(t) = \frac{1}{\epsilon}\{\alpha[f_1(x_1, r_2 + r) - f_{r1}(x_1, r_2)] + A_r r\}$
Perturbation calculation

Figure 2.1: Block diagram of the overall control system

After the first fault mitigation act, (2.2) can be written as,

$$\dot{x}_1 = f_1(x_1, x_2) \tag{2.3a}$$

$$\dot{x}_2 = f_2(x_1, x_2, u_2) \tag{2.3b}$$

With this form, proposed fault mitigation algorithm can be analyzed as a control problem of general interconnected or cascade systems.

The proposed control law can be regarded as introduction of an additional state as a secondary command on the trajectory of $x_2$. This secondary command is generated by a singular dynamic system, whose structure is determined in order to render the $x_1$ dynamics into a desired form. With this control structure, overall closed loop dynamics can be formulated as a singular perturbation problem and results of the Tikhonov's theorem can be used to analyze the behaviour of the system. Block diagram of the proposed control system architecture is shown in Figure 2.1.

The idea of the developed control architecture is to construct a trajectory for $x_2$ as $x_2(t) = r_2(t) + r(t)$ such that $r(t)$ stabilizes the $x_1$ dynamics. This perturbation on the trajectory of $x_2$ is generated by a singular dynamic system, whose structure is determined in order to render the $x_1$ dynamics into a desired form. With this structure, overall closed loop dynamics can be formulated as a singular perturbation problem.

Tikhonov's theorem on singular perturbations of dynamic systems is the main tool for the proposed method. Therefore before the presentation of the proposed control architecture and the design method, Tikhonov's theorem and some definitions and results from the analysis of time-varying and perturbed systems, that will be used in the proof of the main theorem are explained in the next section.

## 2.3 Mathematical Preliminaries

The mathematical foundation of the proposed method relies mainly on Singular Perturbation Theory. Basic definitions and main theorems from the literature are reviewed in this section. First, the main theorem of the Singular Perturbation Theory, also known as the "Tikhonov's Theorem" is explained. Then some definitions and results from the stability analysis of nonlinear time-varying systems are given in the subsequent sections.

### 2.3.1 Tikhonov's Theorem

Singular perturbation problem can be stated by considering the following system of equations, where $\epsilon$ is a small parameter [4]:

$$\begin{aligned} \dot{x} &= f(t, x, z, \epsilon), \qquad x(t_0) = \xi(\epsilon) \\ \epsilon \cdot \dot{z} &= g(t, x, z, \epsilon), \qquad z(t_0) = \eta(\epsilon) \end{aligned} \tag{2.4}$$

**Theorem 2.3.1** (Tikhonov's Theorem)**.** *Consider the singular perturbation problem (2.4) and let $z = h(t, x)$ be an isolated root of the equation $g(t, x, z, 0) = 0$. Assume that the following conditions are satisfied for all*

$$[t, x, z - h(t, x), \epsilon] \in [0, t_1] \times B_r \times B_\rho \times [0, \epsilon_0]$$

- *The functions $f$, $g$ and their first derivatives with respect to $(x, z, \epsilon)$, the function $h(t, x)$ and the Jacobian $\partial g(t, x, z, 0)/\partial z$ and their first partial derivatives with respect to their arguments are continuous. The initial data $\xi(\epsilon)$ and $\eta(\epsilon)$ are smooth functions of $\epsilon$.*
- *The reduced problem $\dot{x} = f(t, x, h(t, x), 0)$, with $x(t_0) = \xi(0)$ has a unique solution $\bar{x}(t)$, defined on $[t_0, t_1]$, and $\|\bar{x}(t)\| \leq r_1 < r$ for all $t \in [t_0, t_1]$.*
- *The origin of the boundary layer model $\frac{dy}{d\tau} = g(t, x, y + h(t, x), 0)$ is exponentially stable, uniformly in $(t, x)$.*

*Then there exists positive constants* $\mu$ *and* $\epsilon^*$ *such that for all* $\|\eta(0) - h(t_0, \xi(0))\| < \mu$ *and* $0 < \epsilon < \epsilon^*$*, the singular perturbation problem (2.4) has a unique solution* $x(t, \epsilon), z(t, \epsilon)$ *on* $t \in [t_0, t_1]$*. With* $\hat{y}(\tau)$ *being the solution of the boundary layer model, following relations hold uniformly for* $t \in [t_0, t_1]$*:*

$$x(t, \epsilon) - \bar{x}(t) = \mathcal{O}(\epsilon)$$
$$z(t, \epsilon) - h(t, \bar{x}(t)) - \hat{y}(t/\epsilon) = \mathcal{O}(\epsilon)$$

Tikhonov's theorem is quite general and applicable also to multidimensional systems. As explained in Section 1.4, it has many applications from system reduction to control system design. Slightly different versions are also available in the literature. Although the stated result is the same, some versions assume asymptotic stability of the reduced problem and the boundary layer model [50, 59], instead of exponential stability. However, it is known that systems where origin is asymptotically, but not exponentially, stable is not robust to smooth perturbations with arbitrary small linear growth bounds [60]. Exponential stability condition assures a robust solution and therefore it is retained in the development of the proposed method.

### 2.3.2 Definitions related to stability of time-varying systems

Following definitions and lemma will be used for proving the stability of the proposed structure. Presented versions are taken from the book "Nonlinear Control" authored by Hassan K. Khalil [61].

**Definition 2.3.1.** Consider the following $n$-dimensional system

$$\dot{x} = f(t, x), \qquad x(t_0) = 0$$

in which $t_0 \geq 0$, $f$ is piecewise continuous in $t$ and locally Lipschitz in $x$ for all $t \geq 0$ and $x \in D$, where $D \subset \mathbb{R}^n$ is domain that contains origin.

The solutions of this system is uniformly bounded if there exists $c > 0$, independent of $t_0$, and for every $a \in (0, c)$, there is $\beta > 0$, dependent on $a$ but independent of $t_0$, such that

$$\|x(t_0)\| \leq a \to \|x(t)\| \leq \beta, \forall t \geq t_0$$

**Definition 2.3.2.** Class $\mathcal{K}$ and $\mathcal{KL}$ functions:

- A scalar continuous function $\alpha(r)$, defined for $r \in [0, a)$, belongs to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$. It belongs to class $K_\infty$ if it is defined for all $r \geq 0$ and $\alpha(r) \to \infty$ as $r \to \infty$.
- A scalar continuous function $\beta(r, s)$, defined for $r \in [0, a)$ and $s \in [0, \infty)$, belongs to class $\mathcal{KL}$ if, for each fixed $s$, the mapping $\beta(r, s)$ belongs to class $\mathcal{K}$ with respect to $r$ and, for each fixed $r$, the mapping $\beta(r, s)$ is decreasing with respect $s$ and $\beta(r, s) \to 0$ as $s \to \infty$.

**Definition 2.3.3.** The system $\dot{x} = f(x, u)$ is input-to-state stable if there exist a class $\mathcal{KL}$ function $\beta$ and a class $\mathcal{K}$ function $\gamma$ such that for any $t_0 \geq 0$, any initial state $x(t_0)$, and any bounded input $u(t)$, the solution $x(t)$ exists for all $t \geq t_0$ and satisfies

$$\|x(t)\| \leq \max \left\{ \beta \left( \|x(t_0)\|, t - t_0 \right), \gamma \left( \sup_{t_0 \leq \tau \leq t} \|u(\tau)\| \right) \right\}, \forall t \geq t_0$$

Input-to-state stability of $\dot{x} = f(x, u)$ implies the following properties:

- For any bounded input $u(t)$, the state $x(t)$ is bounded;
- $x(t)$ is ultimately bounded by $\gamma \left( \sup_{t_0 \leq \tau \leq t} \|u(\tau)\| \right)$;
- if $u(t)$ converges to zero as $t \to \infty$, so does $x(t)$;
- The origin of the unforced system $\dot{x} = f(x, 0)$ is globally asymptotically stable.

Following lemma (Lemma 4.5, p.108 [61]) gives a sufficient condition for input-to-state stability.

**Lemma 2.3.2.** *Suppose $f(x, u)$ is continuously differentiable and globally Lipschitz in $(x, u)$. If $\dot{x} = f(x, 0)$ has a globally exponentially stable equilibrium point at the origin, then the system $\dot{x} = f(x, u)$ is input-to-state stable.*

*Proof.* This lemma is a consequence of converse Lyapunov theorem, that states the existence of a Lyapunov function for exponentially stable systems and its proof can be found in the original sources [4, 61]. □

### 2.3.3 Stability of time-varying and perturbed systems

**Definition 2.3.4** (Exponential stability)**.** Let $f(x)$ be a locally Lipschitz function defined over a domain $D \subset R^n$, which contains the origin and $f(0) = 0$. The equilibrium point $x = 0$ of $\dot{x} = f(x)$ is exponentially stable if there exists positive constants $c, k$ and $\lambda$ such that,

$$\|x(t)\| \leq k \|x(t_0)\| e^{-\lambda(t-t_0)}, \forall t \geq 0$$

for all $\|x(0)\| < c$. It is globally exponentially stable if the inequality is satisfied for every initial state $x(0)$.

This condition actually states that, a trajectory started within a certain region of the phase space is contained within an exponentially decreasing function. Proof of exponential stability is achieved by finding appropriate Lyapunov functions for the system under consideration. Of course, linear stable time invariant systems are exponentially stable with $\lambda$ corresponding to eigenvalues. But for nonlinear systems, there is no general methodology. Some useful results are presented by Khalil and the following results are based on his lecture notes [60].

Following theorem poses a sufficient condition for exponential stability of the origin for a dynamic system.

**Theorem 2.3.3.** *Let the origin $x = 0$ be an equilibrium point of $\dot{x} = f(t, x)$ and $D \subset R^n$ be a domain containing $x = 0$. Suppose $f(t, x)$ is piecewise continuous in t and locally Lipschitz in $x$ for all $t \geq 0$ and $x \in D$. Let $V(t, x)$ be a continuously differentiable function such that*

$$k_1 \|x\|^a \leq V(t, x) \leq k_2 \|x\|^a$$
$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) \leq -k_3 \|x\|^a$$

*for all $t \geq 0$ and $x \in D$, where $k_1, k_2, k_3$ and $a$ are positive constants. Then, the origin is exponentially stable. If the assumptions hold globally, the origin will be globally exponentially stable.*

If the dynamic system is time-invariant, then the corresponding Lyapunov function $V(t, x)$ would be independent of $t$.

A possible approach to the analysis of a time dependent nonlinear system is, treating it the time dependent terms as a perturbation to an exponentially stable, time-invariant nominal system. Following corollaries summarizes the results that can be achieved from such an approach.

**Corollary 2.3.1.** *Consider the perturbed system*

$$\dot{x} = f(x) + g(t,x), \quad g(t,0) = 0$$

*where $\dot{x} = f(x)$ is an exponentially stable system and in accordance with Theorem 2.3.3, let the corresponding Lyapunov function $V(x)$ has the following bounds:*

$$\begin{aligned} c_1 \left\|x\right\|^2 \leq V(x) \leq c_2 \left\|x\right\|^2 \\ \frac{\partial V}{\partial x} f(x) \leq -c_3 \left\|x\right\|^2 \\ \left\|\frac{\partial V}{\partial x}\right\| \leq c_4 \left\|x\right\| \end{aligned}$$

*The origin of the perturbed system is exponentially stable if*

$$\left\|g(t,x)\right\| \leq \gamma \left\|x\right\|, \quad 0 \leq \gamma < \frac{c_3}{c_4}$$

*Proof.* Use the Lyapunov function of the original system $V(x)$ as a Lyapunov function for the perturbed system:

$$\begin{aligned} \dot{V}(t,x) &= \frac{\partial V}{\partial x} f(x) + \frac{\partial V}{\partial x} g(t,x) \\ \Rightarrow \dot{V}(t,x) &\leq \frac{\partial V}{\partial x} f(x) + \left\|\frac{\partial V}{\partial x} g(t,x)\right\| \\ &\leq -c_3 \left\|x\right\|^2 + c_4 \left\|x\right\| \left\|g(t,x)\right\| \end{aligned}$$

if $\left\|g(t,x)\right\| \leq \gamma \left\|x\right\|$, then the last inequality can be written as,

$$\dot{V}(t,x) \leq -c_3 \left\|x\right\|^2 + c_4 \gamma \left\|x\right\|^2 = -\left(c_3 - \gamma c_4\right) \left\|x\right\|^2$$

$c_3/c_4 > \gamma$, together with $\left\|g(t,x)\right\| \leq \gamma \left\|x\right\|$ condition guarantees that $\dot{V}(t,x) < 0$, which indicates the exponential stability of the perturbed system. □

**Corollary 2.3.2.** *For the system in Corollary 2.3.1, if the original system is linear stable system, i.e., $f(x) = Ax$ where $A$ is Hurwitz, that satisfies the Lyapunov equation $PA + A^T P + Q = 0$ with $P$ for an arbitrary $Q = Q^T > 0$, then the origin of the perturbed system is exponentially stable if*

$$\|g(t,x)\| \leq \gamma \|x\|, \quad 0 \leq \gamma < \frac{\lambda_{min}(Q)}{2\lambda_{max}(P)}$$

*Proof.* Considering $V(x) = x^T P x$ as the Lyapunov function for the dynamic system $\dot{x} = Ax$. $P$ is a solution to the Lyapunov equation $PA + A^T P = -Q$ with $Q = Q^T > 0$. Then it can be written that,

$$\lambda_{min}(P) \|x\|^2 \leq V(x) \leq \lambda_{max}(P) \|x\|^2$$

$$\dot{V}(x) = -x^T Q x \leq -\lambda_{min}(Q) \|x\|^2$$

$$\frac{\partial V(x)}{\partial x} = x^T P + x^T P^T \rightarrow \left\| \frac{\partial V(x)}{\partial x} \right\| = \left\| x^T \left(P + P^T\right) \right\| = 2 \|P\| \|x\|$$

$$\Rightarrow \left\| \frac{\partial V(x)}{\partial x} \right\| \leq 2\lambda_{max}(P) \|x\|$$

Therefore, $c_3 = \lambda_{min}(Q)$ and $c_4 = 2\lambda_{max}(P)$. Hence,

$$\gamma < \frac{\lambda_{min}(Q)}{2\lambda_{max}(P)}$$

is a sufficient condition for exponential stability of the perturbed system. □

*Remark* 2.3.1. Since Lyapunov equation can be solved for any $Q = Q^T > 0$ (Theorem 2.3.5), it is better to choose $Q$ such that the ratio $\frac{\lambda_{min}(Q)}{2\lambda_{max}(P)}$ is maximized. It can be shown that this ratio is maximized for $Q$ being the identity matrix [4].

### 2.3.4 Existence of solutions

For a general nonlinear dynamic system $\dot{x} = f(x,t)$, Cauchy existence theorem states that continuity of $f(x,t)$ is sufficient for local existence of a solution, however for the uniqueness of the solution, $f(t,x)$ should also satisfy Lipschitz condition:

**Theorem 2.3.4.** *If the function $f(x,t)$ is continuous in t, and if there exists a strictly constant L such that*

$$\|f(x_2,t) - f(x_1,t)\| \leq L \|x_2 - x_1\| \tag{2.5}$$

*for all $x_1$ and $x_2$ in a finite neighborhood of the origin and all $t$ in the interval $[t_0, t_0 + T]$ (with $T$ being a strictly positive constant), then $\dot{x} = f(x, t)$ has a unique solution $x(t)$ for sufficiently small initial states and in a sufficiently short time interval.*

Condition (2.5) is called the Lipschitz condition and it is sufficient for uniqueness of the solution. Continuous functions with bounded first derivatives (within the domain of interest) satisfy Lipschitz condition. It should be noted that Lipschitz condition is a rather conservative condition and for most physical problems, smoothness of $f$ is enough for the existence and uniqueness of the solution of the dynamic system [15].

Lyapunov equation, which can be written as $PA + A^T P = -Q$, is widely used in control system design methods and the following theorem (Theorem 3.7, p.72 [61]) provides the existence of the solution of Lyapunov's equation.

**Theorem 2.3.5.** *A matrix $A$ is Hurwitz if and only if for every positive definite symmetric matrix $Q$ there exists a positive definite symmetric matrix $P$ that satisfies the Lyapunov equation $PA + A^T P = -Q$. Moreover, if $A$ is Hurwitz, then $P$ is the unique solution.*

*Remark* 2.3.2. Theorem 2.3.5 is biconditional. Therefore, negative definiteness of $A$ ("Hurwitz Matrix") implies the existence of the solution of the Lyapunov equation.

## 2.4 Proposed Control Architecture

Mathematical foundations of the proposed method can be laid down with considering the following dynamic system,

$$\Sigma_0 : \quad \begin{aligned} \dot{x}_1 &= f_1(x_1, x_2) && \text{(2.6a)} \\ \dot{x}_2 &= f_2(x_1, x_2, u_2) && \text{(2.6b)} \end{aligned}$$

where $x_1 \in \mathbb{R}^n$, $x_2 \in \mathbb{R}^m$, $u \in \mathbb{R}^m$.

Following theorem provides the necessary conditions for achieving the stability of $x_1$ around origin, while $x_2$ can be maintained within a bounded vicinity of a time dependent reference trajectory $r_2(t) \in \mathbb{R}^m$ using the proposed method.

**Theorem 2.4.1.** *The interconnected system* $\Sigma_0$ *can be stabilized around the trajectory* $x_2(t) = r_2(t) + Br$ *such that there exists* $c > 0$*, independent of* $t_0$*, and for every* $a \in (0, c)$*, there is* $\beta > 0$*, dependent on* $a$ *but independent of* $t_0$*, such that,*

$$\|x_1(t_0)\| \leq a \rightarrow \|x_1(t)\| \leq \beta, \forall t \leq t_0$$

*and* $\|x_2(t) - r_2(t)\| < \beta_2$*, where* $r \in \mathbb{R}^n$ *is the output of a singularly perturbed system,*

$$\epsilon \cdot \dot{r} = \alpha \cdot [f_1(x_1, r_2 + Br) - f_{r1}(x_1, r_2)] + A_r r \tag{2.7}$$

*with* $\epsilon$ *as a small parameter and* $\alpha = \pm 1$*.* $f_{r1}(x_1, x_2)$ *is a stable reference model with origin as a stable equilibrium point and* $A_r$ *is a Hurwitz matrix,* $B$ *is a constant matrix with dimensions* $B \in \mathbb{R}^{n \times m}$*, provided that the following assumptions hold,*

*A1 The functions* $f_1$ *and* $f_2$ *are at least locally Lipschitz functions with continuous and bounded derivatives and* $f_1(0, 0) = 0$*.*

*A2 The homogeneous system (2.6a), with* $x_2$ *as the control input, is small-time locally controllable from* $x_{1,0} = 0$*,* $\forall x_2 \in \mathbb{R}^m : x_2 \subset \Omega$*.*

*A3 The homogeneous system (2.6b) is small-time locally controllable from* $x_{2,0} = 0$*,* $\forall x_1 \in \mathbb{R}^n : x_1 \subset \Omega$ *and there exists a smooth feedback control system that provides control inputs* $u = u(x_1, x_2)$ *such that error between the reference command* $x_{2c}$ *and* $x_2$ *converges to zero:*

$$\lim_{t \to \infty} e_2 = \lim_{t \to \infty} \|x_2 - x_{2c}\| = 0$$

*A4 Internal dynamics* $f_1(x_1, x_2)$ *has the following bound:*

$$\|f_1(0, x_2)\| \leq \gamma_2 \|x_2\|$$

*and the perturbed system satisfies the following conditions,*

*C1* $A_r$ *satisfies the following bound,*

$$\left\| \left. \frac{\partial f_1}{\partial x_2} \right|_{[x_1(t), r_2]} B \right\| \leq \gamma_1 \qquad \textit{and} \qquad \gamma_1 < \frac{1}{2\lambda_{max}(P)}$$

*where* $P$ *is the solution to the Lyapunov equation for* $A_r$*:* $PA_r + A_r^T P + I = 0$*.*

*C2 $A_r$ satisfies the following bound,*

$$\|f_1(0, x_2)\| \leq \gamma_2 \|x_2\| \qquad \textit{and} \qquad \gamma_2 < \frac{1}{2\lambda_{max}(P)}$$

*where $P$ is the solution to the Lyapunov equation for $A_r$: $PA_r + A_r^T P + I = 0$.*

*Proof.* The theorem will be proved by first showing that the system of equations (2.6) and (2.7) is a singular perturbation problem and the conditions of the Tikhonov's theorem will be verified. The boundedness of $x_1$ will be shown using the results of the Tikhonov's theorem and boundedness of $r(t)$ will be proved by showing that $r(t)$ dynamics is input-to-state stable, under the provided assumptions.

1. The system of equations (2.6) and (2.7) can be formulated as a singular perturbation problem by observing that for $x_{c2} = r_2 + Br$, existence of a stable control system (Assumption *A3*) guarantees that the error signal $e_2 = x_2 - (r_2 + Br)$ is uniformly bounded such that $\|e_2(t)\| \leq c$. The error dynamics $\dot{e}_2$ can be considered as a stable dynamic system $g_2$ as,

   $$\dot{e}_2 = g_2(x_1, e_2, u)$$

   With this representation, $e_2$ can be used as the state variable, instead of $x_2$. This change of variables leads to the following system.

   $$\dot{x}_1 = f_1(x_1, e_2 + r_2 + r) \tag{2.8a}$$

   $$\dot{e}_2 = g_2(x_1, e_2, u) \tag{2.8b}$$

   $$\epsilon \cdot \dot{r} = \alpha \cdot [f_1(x_1, e_2 + r_2 + Br) - f_{r1}(x_1, r_2)] + A_r r \tag{2.8c}$$

   For $\epsilon$ is a small parameter, interconnected system (2.8) can be regarded as a singular perturbation problem. Reduced problem of the singular perturbation problem (2.8) is obtained with setting $\epsilon$ to zero. Since neither $x_1$ dynamics, nor $e_2$ dynamics involve $\epsilon$ term explicitly, reduced problem is equivalent to the original uncontrolled system $\Sigma_0$:

   $$\begin{aligned} \dot{x}_1 &= f_1(x_1, e_2 + r_2 + Br) \\ \dot{e}_2 &= g_2(x_1, e_2, u) \end{aligned} \tag{2.9}$$

The boundary layer model of the singular perturbation problem (2.8) can be obtained with replacing $r$ with $y + h(x_1)$ in $\epsilon\dot{r}$ equation:

$$\frac{dy}{d\tau} = \alpha \cdot [f_1(x_1, r_2 + By + Bh(x_1)) - f_{r1}(x_1, r_2)] + A_r(y + h(x_1)) \quad (2.10)$$

where $h(x_1)$ is a solution of the equation $\alpha \cdot [f_1(x_1, r_2 + Bh(x_1)) - f_{r1}(x_1, r_2)] + A_r h(x_1) = 0$ and $\tau = t/\epsilon$.

Before using the results of the Tikhonov's theorem, its assumptions should be verified.

2. First assumption of the Tikhonov's theorem (*A1* of Theorem 2.3.1) is related to smoothness of the dynamic system and it is satisfied for smooth $f_1, f_2, g_2, f_{r1}$ functions. This assumption is also retained in this work (Assumptions *A1* and *A3*).

3. Second assumption of the Tikhonov's theorem (*A2* of Theorem 2.3.1) is related to the uniqueness of the solution of the reduced problem (2.9).

   Since it is assumed that both $f_1$ and $g_2$ are continuous and bounded (Assumptions *A1* and *A3*), existence and uniqueness of the solution of this system van be guaranteed using the Cauchy existence theorem (Theorem 2.3.4).

4. Third assumption of the Tikhonov's theorem (*A3* of Theorem 2.3.1) requires that origin is an exponentially stable equilibrium point of the boundary layer model (2.10).

   Derivation of global exponential stability conditions using Definition 2.3.4 and Theorem 2.3.3 is difficulty and highly system specific. Therefore, conditions of local exponential stability will be sought, using Taylor series expansion of $f_1$.

   Since boundary layer equation represents the deviation dynamics from the solution of the reduced problem, it is reasonable to expand $f_1$ around this solution

$$f_1(x_1, x_2) = f_1(x_1(t), \bar{x}_2(t)) + \left.\frac{\partial f_1}{\partial x_2}\right|_{[x_1(t), \bar{x}_2(t)]} [x_2(t) - \bar{x}_2(t)]$$

   For $\bar{x}_2(t) = r_2$, boundary layer equation (2.10) can be rewritten as;

$$\begin{aligned} \frac{dy}{d\tau} = \alpha \cdot &\left[ f_1(x_1, r_2) + \left.\frac{\partial f_1}{\partial x_2}\right|_{[x_1(t), r_2]} (By + Bh(x_1)) - f_{r1}(x_1, r_2) \right] \\ &+ A_r y + A_r h(x_1)) \end{aligned} \quad (2.11)$$

By definition, $h(x_1)$ is an equilibrium solution of the perturbation equation $\epsilon\dot{r} = 0$ and hence,

$$\alpha \cdot \left[ f_1(x_1, r_2) + \left.\frac{\partial f_1}{\partial x_2}\right|_{[x_1(t), r_2]} Bh(x_1) - f_{r1}(x_1, r_2) \right] + A_r y + A_r h(x_1) = 0$$

Removing the corresponding terms from the (2.11) results in,

$$\frac{dy}{d\tau} = \left( A_r + \alpha \cdot \left.\frac{\partial f_1}{\partial x_2}\right|_{[x_1(t), r_2]B} \right) y \tag{2.12}$$

Boundary layer model (2.12) is in the form of a perturbed linear time varying system. For $\alpha \cdot \left.\frac{\partial f_1}{\partial x_2}\right|_{[x_1(t), r_2]} B < 0$, boundary layer equation is always exponentially stable. But if this is not the case, results of the Corollary 2.3.2 can be used for derivation of sufficient conditions of exponential stability.

Existence of a bound on the derivative of $f_1$ is assured in Assumption *A1* and therefore exponentially stability of the boundary layer model (2.12) can be guaranteed if,

$$\left\| \left.\frac{\partial f_1}{\partial x_2}\right|_{[x_1(t), r_2]} B \right\| \leq \gamma_1 \qquad \text{and} \qquad \gamma_1 < \frac{\lambda_{min}(Q)}{2\lambda_{max}(P)}$$

where $P$ is the solution to the Lyapunov equation for $A_r$: $PA_r + A_r^T P + Q = 0$, $Q = Q^T > 0$ and existence of a stable matrix $A_r$ with the desired bound can be verified using Theorem 2.3.5 and Remark 2.3.2. Using identity matrix as $Q$ (see Remark 2.3.1) results in the condition given in *C1* of Theorem 2.4.1.

5. With all assumptions of the Tikhonov's theorem, Theorem 2.3.1 is satisfied, its results can be used to show that for $x = (x_1, e_2)$ in (2.6),

$$x(t) - \bar{x}(t) = \mathcal{O}(\epsilon)$$

where $\bar{x}(t)$ is the solution of the reduced problem (2.9).

The order of magnitude notation can be replaced with an inequality,

$$x(t) - \bar{x}(t) = \mathcal{O}(\epsilon) \qquad \rightarrow \qquad |x(t) - \bar{x}(t)| \leq K \cdot \epsilon$$

such that there exists a positive number $K$ for all $t \in [t_0, t_1]$.

For $\epsilon = 0$, the perturbed system (2.7) has the solution,

$$f_1(x_1, x_2) = f_{r1}(x_1, r_2)$$

and therefore $\bar{x}(t) = (\bar{x}_1(t), \bar{e}_2(t))$ is the solution of the dynamic system,

$$\begin{aligned} \dot{\bar{x}}_1 &= f_{r1}(\bar{x}_1, r_2) \\ \dot{\bar{e}}_2 &= g_2\left(\bar{x}_1, \bar{e}_2, u\right) \end{aligned}$$

Since $f_{r1}$ is a stable reference model for $x_1$ dynamics, it is bounded. Let this norm be $\|\bar{x}_1(t)\| < \bar{\beta}$. Also, since $x_2$ error dynamics $e_2$ is stable, an upper bound can be found such that $\|\bar{e}_2\| \leq \bar{\beta}_e$. For $\bar{x}(t) = (\bar{x}_1(t), e_2(t))$, it can be concluded that,

$$\|\bar{x}(t)\| \leq \bar{\beta} + \bar{\beta}_e$$

Then the existence of an upper bound on $\|x_1(t)\|$ can be shown using several triangular inequalities for norms:

$$\begin{aligned} |\|x_1(t)\| - \|\bar{x}_1(t)\|| &\leq |x_1(t) - \bar{x}_1(t)| \\ \text{Since, } |x_1(t) - \bar{x}_1(t)| &\leq K \cdot \epsilon \rightarrow |\|x_1(t)\| - \|\bar{x}_1(t)\|| \leq K \cdot \epsilon \\ \|x_1(t)\| &\leq \|\bar{x}_1(t)\| + K \cdot \epsilon \\ \text{Hence, } \|x_1(t)\| &\leq \bar{\beta} + \bar{\beta}_e + K \cdot \epsilon \end{aligned}$$

This completes the proof of the main result that,

$$\|x_1(t_0)\| \leq a \rightarrow \|x_1(t)\| \leq \beta, \forall t \leq t_0$$

with $\beta = \bar{\beta} + \bar{\beta}_e + K \cdot \epsilon$. This relation on state bounds is shown schematically in Figure 2.2.

6. The final thing that should be considered is the ultimate boundedness of the perturbation signal $r(t)$. This can be proved if it can be shown that $r(t)$ dynamics (2.7) is input-to-state stable (Definition 2.3.3). Lemma 2.3.2 can be used for this purpose, where it is stated that exponential stability of $\dot{x} = f(x, 0)$ is sufficient for input-to-state stability of $\dot{x} = f(x, 0)$.

   $x_1$ and $r_2$ can be regarded as external inputs in (2.7). For $x_1(t), r_2(t) = 0$, this equation can be written as,

$$\epsilon \cdot \dot{r} = \alpha \cdot [f_1(0, r) - f_{r1}(0, 0)] + A_r r$$

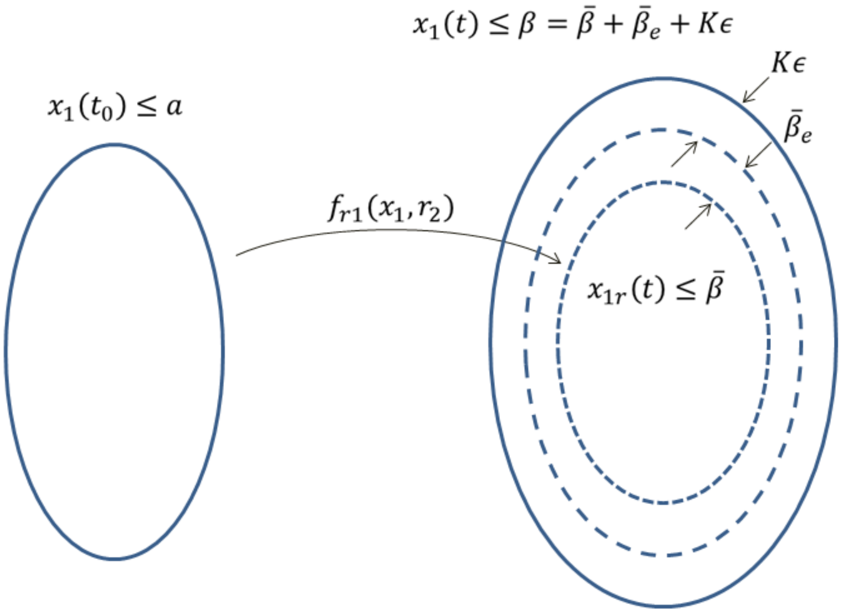

$x_1(t) \leq \beta = \bar{\beta} + \bar{\beta}_e + K\epsilon$
$K\epsilon$
$x_1(t_0) \leq a$
$\bar{\beta}_e$
$f_{r1}(x_1, r_2)$
$x_{1r}(t) \leq \bar{\beta}$

Figure 2.2: Schematic view of the bound on the states $x_1$.

Since $f_{r1}(0,0) = 0$ (Assumption *A1*), resultant system can be written as,

$$\epsilon \cdot \dot{r} = \alpha f_1(0, r) + A_r r \tag{2.13}$$

$\epsilon$ is a positive constant and therefore it does not effect the stability condition. Since $A_r$ is Hurwitz and a design parameter, Corollary 2.3.1 can be used.

Since $\alpha = \pm 1$ and $\|f_1(0, r)\| \leq \gamma_2 \|r\|$ (Assumption *A4*), (2.13) is exponentially stable provided that $A_r$ satisfies the bound:

$$\|f_1(0, r)\| \leq \gamma_2 \|r\| \qquad \text{and} \qquad \gamma_2 < \frac{1}{2\lambda_{max}(P)}$$

where $P$ is the solution to the Lyapunov equation for $A_r$: $PA_r + A_r^T P + I = 0$. This is the condition given in *C2* of Theorem 2.4.1.

With this condition, input-to-state stability of $r(t)$ dynamics can be guaranteed and, as stated in Definition 2.3.3, upper bound on $r(t)$ can be written as,

$$r(t) \leq \gamma \left( \sup_{t_0 \leq \tau \leq t} (\|x_1(\tau)\| + \|r_2(\tau)\|) \right)$$

This completes the proof.

$\square$

*Remark* 2.4.1. Tikhonov's theorem provides a result where convergence on the solution of the reduced problem is achieved in order of magnitude basis. Since $K < \infty$, if $\epsilon \to 0$ then $K \cdot \epsilon \to 0$ and theoretically the system would converge to the reduced problem solution. However, depending on the value of $\epsilon$ and initial conditions, $K$ can be large and this might result in larger $x_1$ and $r$ values, which might exceed the system limits in practice.

*Remark* 2.4.2. Conditions for exponential stability of the origin of the boundary layer equation (2.12) is provided only for the linearized form of it. Validity of this linearization also depends on initial conditions and deviations from the desired behaviour, which is characterized by the reduced problem.

*Remark* 2.4.3. Tikhonov's theorem provides a limiting behaviour. Transient dynamics such as peak response or convergence time of the overall system is affected by the choice of $\epsilon$, $f_{r1}$ and $x_2$ controller, through $e_2$ dynamics. These aspects of the control system are more problem specific and therefore this aspect of the problem is discussed at example applications.

*Remark* 2.4.4. $A_r$, $f_1(r)$ and $\epsilon$ are design parameters that should be adjusted within the design process. System specific conditions *C1* and *C2* provides bounds on $A_r$. Existence of such an $A_r$ that satisfies any desired bound is guaranteed with Theorem 2.3.5 and Remark 2.3.2

*Remark* 2.4.5. Constant matrix $B \in \mathbb{R}^{m \times n}$ is used to project the calculated perturbation $r \in \mathbb{R}^n$ to $x_2$ trajectory which is $\mathbb{R}^m$. For $m = n$, identity matrix is a natural choice but for the cases where $m \neq n$, this matrix determines how the calculated perturbations are distributed to the directly controlled states. This usage is explained though an example in Section 2.5.2.

*Remark* 2.4.6. The theorem is developed for systems where origin is an equilibrium point of the internal dynamics, *i.e.*, $f_1(0,0), f_{r1}(0,0) = 0$. However, it is actually merely a convenience and the proofs are still valid for any other equilibrium point as well, since a shift in variables would result in a state variable set where origin is an equilibrium. Such kind of situations occurs in for example partial failure cases, where a residual control force still remains. This is automatically handled by the algorithm by shifting the equilibrium point where the net resultant force becomes zero. Such a case is discussed in Chapter 4.

However for some systems, it is not possible to nullify such an effect through a shift in the state variables. For example consider the system $\dot{x} = \sin x + 2$, which does not have any equilibrium point.

*Remark* 2.4.7. A state estimator based disturbance estimator is developed in Section 2.6. Apart from fault detection purposes, calculated disturbance vector $\hat{d}$ can be used within the perturbation calculation as well:

$$\epsilon \cdot \dot{r} = \alpha \cdot \Big[ f_1(x_1, r_2 + r) + \hat{d} - f_{r1}(x_1, r_2) \Big] + A_r r$$

With such an adaptive term, developed structure can be made robust to modeling errors.

Theorem 2.4.1 includes the conditions and results of the proposed fault tolerant control method. Fault detection and adaptive compensation part will be explained in Section 2.6 and after that, overall design methodology for the proposed architecture will be explained in Section 2.7. But before that, capabilities of the proposed control system will be demonstrated through numerical examples.

## 2.5 Numerical examples

In order to explain the methodology more thoroughly and show how the mathematical relations given in Theorem 2.4.1 can be applied, analytically tractable prototype problems are presented in this section. Numerical examples are chosen in order to reflect important characteristics of the nonlinear control structure that will be used as the fault mitigation algorithm. Fault detection is not addressed in this section, since its implementation is straightforward and system specific remarks are given in corresponding chapters, where application on more complex problems are demonstrated.

### 2.5.1 Example 1

Consider the following dynamical system:

$$\begin{aligned} \dot{x}_1 &= x_1^2 + x_2^3 \\ \dot{x}_2 &= u \end{aligned} \tag{2.14}$$

Typical approach for designing a control system for nonlinear systems is linearization. For $x = \bar{x} + x'$, this results in,

$$\frac{d}{dt}\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} 2\bar{x}_1 & 3\bar{x}_2^2 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

As can be directly seen from the system matrix, the linearized system is uncontrollable around the origin, *i.e.*, $(\bar{x}_1, \bar{x}_2) = (0, 0)$. Therefore it is not possible to design a linear controller that would stabilize the system around the origin.

Furthermore, the zero dynamics of the system ($\dot{x}_1 = f_1(x_1, 0)$) is unstable. A possible approach would be finding a sliding surface with $x_2^3 + x_1^2 - A_{rm1}x_1 = 0$, that can be used with a Sliding Mode Controller or finding a suitable control Lyapunov function. But it is not straightforward to find a smooth nonlinear controller. Proposed method would provide an alternative.

For the stabilization of $x_1$ around the origin and command tracking of $x_2(t)$, consider the following closed loop system.

$$\begin{aligned} \dot{x}_1 &= x_1^2 + x_2^3 \\ \dot{x}_2 &= K \cdot (r_2 + r - x_2) \\ \epsilon\dot{r} &= \alpha \cdot \left[x_1^2 + (r_2 + r)^3 - A_{rm1}x_1\right] + A_r r \end{aligned} \tag{2.15}$$

With comparison to (2.6), $f_1(x_1, x_2) = x_1^2 + x_2^3$ and $f_2(x_1, x_2, u) = u$. $B$ is taken as $1$.

Assumptions *A1* - *A3* of Theorem 2.4.1 are straightforward to verify. For Assumption *A4*, linear growth bound on $f_1(0, x_2) = x_2^3$ should be found:

$$\|f_1(0, x_2)\| = \left\|x_2^2 \cdot x_2\right\| \leq |x_2|^2 \|x_2\|$$

Although $\lim_{x_2 \to \infty} x_2^2 = \infty$, a subset of $\mathbb{R}$ can be chosen for the calculation of an upper bound. For example, for $|x_2| \leq \delta$, $\gamma_2$ in *C1* of Theorem 2.4.1 can be found as $\gamma_2 = \delta^2$. Such kind of a bound can always be found for functions that satisfy Lipschitz condition.

With reference to (2.12), boundary layer equation can be written as,

$$\frac{dy}{d\tau} = \left[A_r + \alpha 3 r_2(t)^2\right] y \tag{2.16}$$

Choosing $\alpha = -1$ is advantageous, since $r_2(t)^2$ term is always positive. Therefore system is stable for any value of $A_r$. Assume that the domain of interest is $x_1, x_2 \in [0, 1]$. Then $\|f_1(0, x_2)\| \leq \gamma_2 \|x_2\|$ with $\gamma_2 = 1$.

Control system (2.15) is implemented with $\alpha = -1$, $K = -10$ and $A_{rm1} = -2$. $A_r$ is taken as $A_r = 0$.

Stabilization around the origin is demonstrated for the case with initial condition of $(x_1, x_2) = (1, 1)$. Response of the system is shown in Figure 2.4. System is brought in to the close vicinity of zero but not exactly zero. Since linearized system is uncontrollable at exactly $x_2 = 0$, any trajectory that misses the origin leaves a residue that results in a very slowly decaying oscillation. Further more, if the trajectory crosses the equilibrium line $x_1 = -x_2^{3/2}$, stops at this point. The value of $A_r$ determines this trajectory and this behaviour is shown in Figure 2.3.

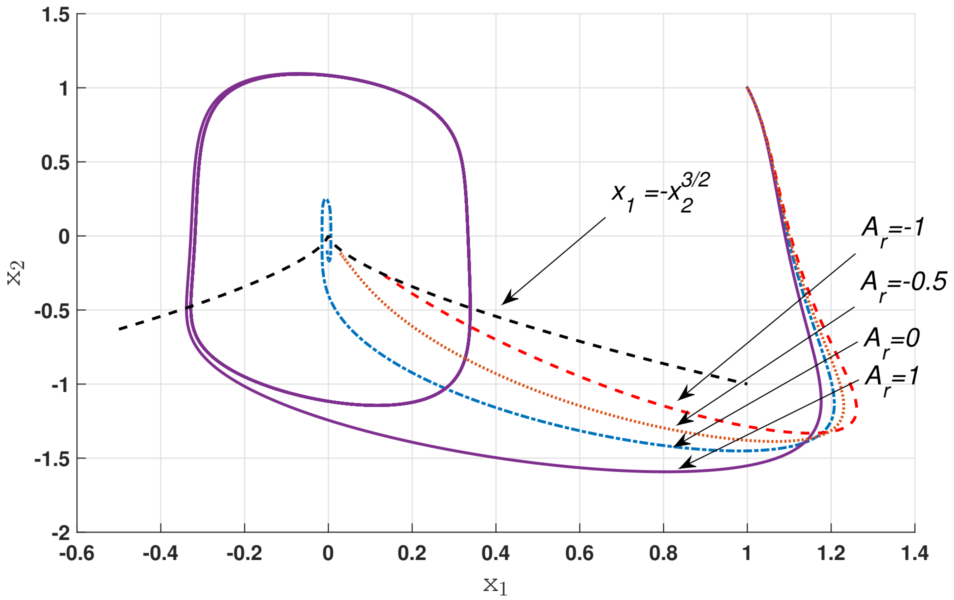


Figure 2.3: Phase space trajectories of the example problem (2.15) with initial condition $(x_1, x_2) = (1, 1)$ for different $A_r$ values.

Sinusoidal trajectory tracking can also be achieved with the same controller parameters. Response of the system is shown in Figure 2.5. Generated perturbation on the nominal trajectory of the system is also shown on Figure 2.6.

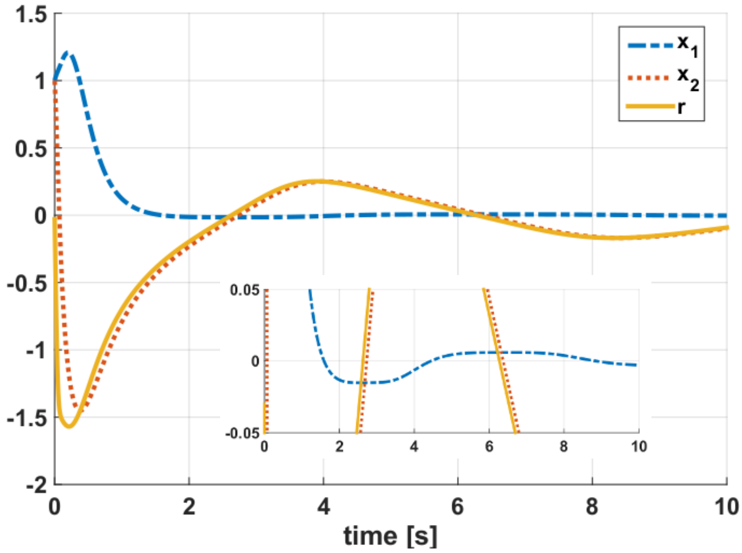


Figure 2.4: Stabilization around the origin for the Example Problem 1 (2.15) with initial condition $(x_1, x_2) = (1, 1)$. System parameters are: $A_r = 0$, $A_{rm1} = -2$ and $K = -10$.

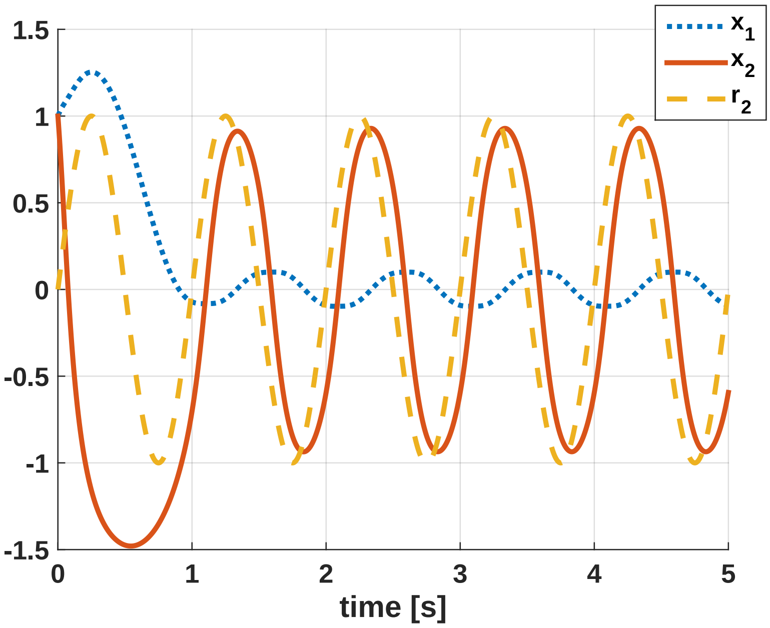


Figure 2.5: Trajectory tracking performance for the Example Problem 1 (2.15) with initial condition $(x_1, x_2) = (1, 1)$ and reference input of $r_2 = sin(2\pi t)$. System parameters are: $A_r = 0$, $A_{rm1} = -2$ and $K = -10$.

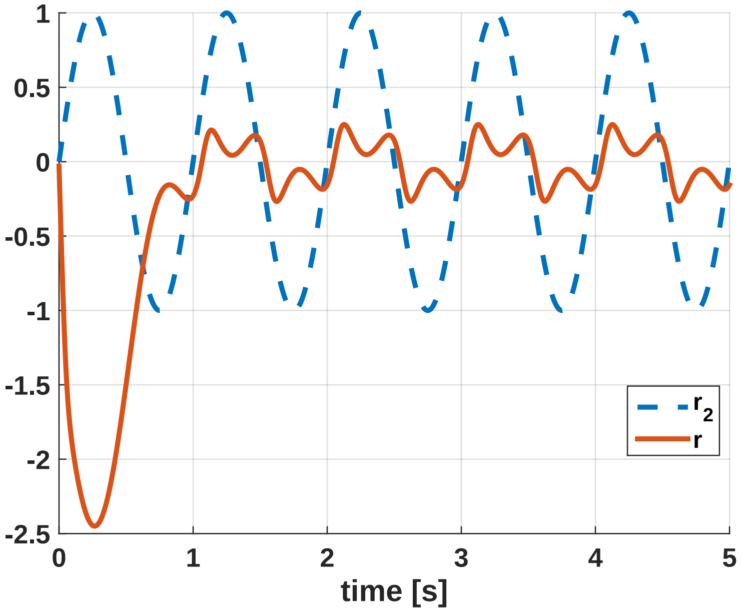


Figure 2.6: Trajectory tracking performance for the Example Problem 1 (2.15) with initial condition $(x_1, x_2) = (1, 1)$ and reference input of $r_2 = sin(2\pi t)$ - Reference input and generated perturbation. System parameters are: $A_r = 0$, $A_{rm1} = -2$ and $K = -10$.

### 2.5.2 Example 2

Attitude control in three dimensions is integral part of any aerospace vehicle's flight control system. A typical fault tolerant control problem in this direction is discussed in Chapter 4. Basic model of this problem that does not include details of the flight dynamics, which is more suitable for theoretical discussions, is analyzed in this section.

Consider the angular momentum equation,

$$I\dot{\omega} = -\omega \times I\omega + M$$

where $\omega \in \mathbb{R}^3$ is the angular velocity vector, $I$ is the inertial moment tensor and $M \in \mathbb{R}$ is the control torque vector.

It is known that angular momentum equation can be stabilized around the origin with a smooth feedback law, even for $M = \begin{bmatrix} 0 & M_1 & M_2 \end{bmatrix}^T$ case [62]. In order to demonstrate the capability of the proposed control architecture, consider the stabilization of the angular momentum of the system around origin, using only a single torque along the axis parallel to $w_3$.

For diagonal $I$, above equation can be written as,

$$\begin{aligned}\dot{q}_1 &= aq_2q_3 \\ \dot{q}_2 &= bq_1q_3 \\ \dot{q}_3 &= cq_1q_2 + u\end{aligned} \tag{2.17}$$

where $\omega = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T$, $a = (I_2 - I_3)/I_1$, $b = (-I_1 + I_3)/I_2$, $c = (I_1 - I_2)/I_3$ and $u = M_3/I_3$. Assume that $I_1 < I_2 < I_3$ so that $a, c < 0$ and $b > 0$.

With comparison to (2.3), states can be grouped as $x_1 = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^T$ and $x_2 = q_3$ with,

$$f_1(x_1, x_2) = \begin{bmatrix} aq_2 \\ bq_1 \end{bmatrix} q_3 \qquad f_2(x_1, x_2, u) = cq_1q_2 + u$$

There are 2 states to be additionally stabilized, but there is only one channel that the perturbation signal can be superposed. One alternative that can be applied in this case is generating two dimensional perturbation signal ($r \in R^2$) and applying the summation of the signals as the stabilizing perturbation: $r \to Br$ with $B = \begin{bmatrix} 1 & 1 \end{bmatrix}$. The closed loop system can be written as,

$$\begin{aligned}\dot{q}_1 &= aq_2q_3 \\ \dot{q}_2 &= bq_1q_3 \\ \dot{q}_3 &= cq_1q_2 + u \\ u &= -cq_1q_2 + K \cdot (r_2 + Br - q_3) \\ \epsilon\dot{r} &= \alpha \left( \begin{bmatrix} 0 & a \\ b & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} (r_2 + Br) - A_{rm1} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \right) + A_r r\end{aligned} \tag{2.18}$$

Linearization term $-cq_1q_2$ is added to the control input, in order to linearize $q_3$ dynamics. Since $a$ and $b$ have opposite signs, choice of $\alpha$ does not matter. $\alpha = 1$ can be chosen for simplicity.

For the determination of $A_r$, consider the Condition *C1* of Theorem 2.4.1 for the bound on the Jacobian of $f_1$:

$$\frac{\partial f_1}{\partial x_2}B = \begin{bmatrix} aq_2 \\ bq_1 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} aq_2 & aq_2 \\ bq_1 & b_1 \end{bmatrix}$$

The upper bound on $\left\| \frac{\partial f_1}{\partial x_2}B \right\|$ can be calculated with using determinant as the norm:

$$\left\| \frac{\partial f_1}{\partial x_2}B \right\| = 0$$

Therefore any positive number $\gamma_1$ can be used as the upper bound.

The Condition *C2* of Theorem 2.4.1 can be formulated similarly,

$$\|f_1(0, x_2)\| = 0$$

Therefore, any stable $A_r$ is sufficient for exponential stability of the boundary layer equation.

For the convergence behaviour of the system, consider the solution of the perturbation equation $\epsilon \dot{r} = 0$:

$$\begin{bmatrix} 0 & a \\ b & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} (r_2 + Br) - A_{rm1} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} + A_r r = 0$$

or with rearrangement of the terms,

$$\left( r_2 \begin{bmatrix} 0 & a \\ b & 0 \end{bmatrix} - A_{rm1} \right) \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} + \left( \begin{bmatrix} 0 & a \\ b & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} B + A_r \right) r = 0$$

Using $B = [1 \quad 1]$, solution of the above equation can be written as,

$$r^*(t) = \frac{1}{bq_1 + aq_2 + A_r} \left( r_2 \begin{bmatrix} 0 & a \\ b & 0 \end{bmatrix} - A_{rm1} \right) \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

In order to achieve bounded $\|r(t)\|$, $A = \left( r_2 \begin{bmatrix} 0 & a \\ b & 0 \end{bmatrix} - A_{rm1} \right) \begin{bmatrix} q_1 & q_2 \end{bmatrix}$ should be a Hurwitz matrix. But then any $(q_1, q_2)$ pair with $\lambda(A)_1 q_1 + \lambda(A)_2 q_2$ is a solution to the equation $r^*(t) = 0$. The situation is depicted in Figure 2.7. In this figure, closed loop response of the system with initial condition $(q_1, q_2, q_3) = (0.1, 0.1, 0.1)$ is shown. Following set of parameters is used in the simulation:

$$a = -1 \quad b = 1 \quad c = -1$$
$$K = -2 \quad A_r = -5 \quad \alpha = 1$$
$$A_{rm1} = \begin{bmatrix} -10 & 0 \\ 0 & -10 \end{bmatrix}$$

Since the reference model $A_{rm1}$ is diagonal with a single eigenvalue of $\lambda_{1,2} = -10$, system converges to the equilibrium point of $(q_1^*, q_2^*, q_3^*) = (-0.1, 0.1, 0)$.

Converge point of the system can be adjusted using non-diagonal reference models. One such configuration is the following parameter set:

$$a = -1 \quad b = 1 \quad c = -1$$
$$K = -2 \quad A_r = -5 \quad \alpha = 1$$
$$A_{rm1} = \begin{bmatrix} -20 & 10 \\ 10 & -10 \end{bmatrix}$$

With this reference model, $q_1$ dynamics is adjusted so that it is faster than the $q_2$ dynamics. Therefore $q_1$ converges to zero, together with $q_3$, while $q_2$ converges to a constant value. The response of the system is shown in Figure 2.8.

Considering the system (2.17), any two states approaching zero makes the third dynamics zero. Therefore, nonzero state stuck at its final position, when other two states converge to origin. But this example demonstrates that it is possible to bound the attitude rates of the system, even with using a single actuator.

### 2.6 Adaptive state estimator as a fault diagnosis algorithm

Fault detection and identification is another critical element of a Fault Tolerant Control structure. As explained in Chapter 1, fault detection algorithm can be designed separately. It is even possible to design different algorithms for first detection and then for identification of a fault. But since the developed control structure involves a system

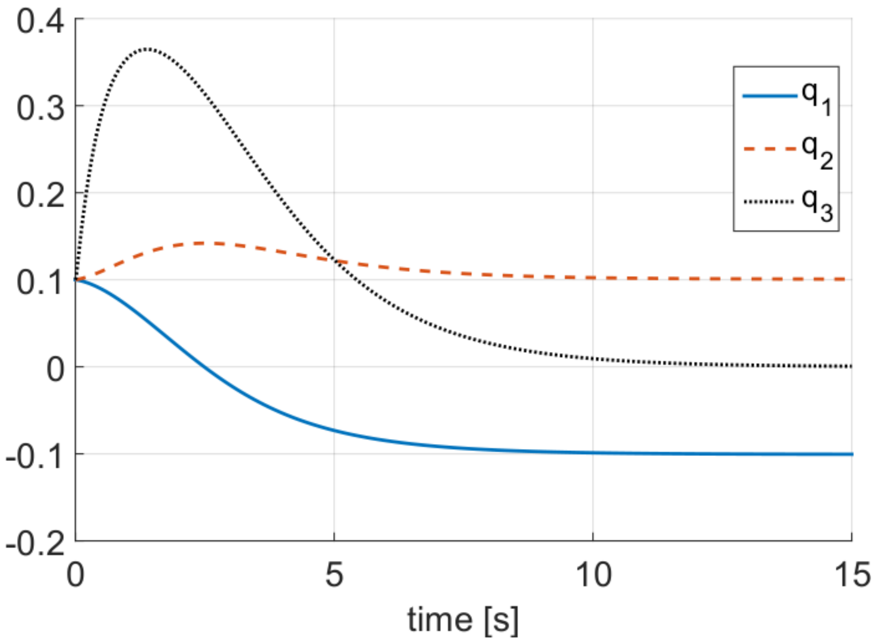


Figure 2.7: Stabilization around the origin for the Example Problem 2 with a diagonal reference model- System parameters are: $a = -1, b = 1, c = -1, K = 2, A_r = -5, \alpha = 1, A_{rm1} = \begin{bmatrix} -10 & 0 \\ 0 & -10 \end{bmatrix}$

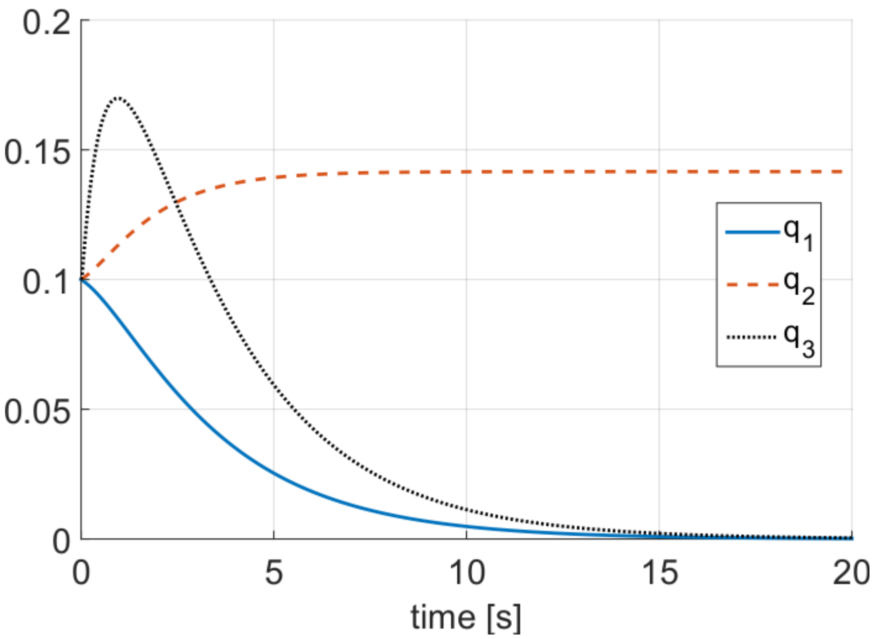


Figure 2.8: Stabilization around the origin for the Example Problem 2 with a non-diagonal reference model - System parameters are: $a = -1, b = 1, c = -1, K = 2, A_r = -5, \alpha = 1, A_{rm1} = \begin{bmatrix} -20 & 10 \\ 10 & -10 \end{bmatrix}$

model, adaptive parts can be added that can be used for fault detection and (in some cases) identification purposes.

State observers are typically used in control systems in order to estimate the states of a system using the discrepancy between the measured outputs and the outputs of the system model. While the original structure is developed by Luenberger [63] for linear systems (hence the name Luenberger Type Observer), Ciccarella, Mora and Germani extended it to nonlinear cases [64].

Consider a general nonlinear dynamic system with $x(t) \in \mathbb{R}^n$:

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}$$

State observer structure can be written for this system as [64],

$$\dot{\hat{x}}(t) = f(\hat{x}(t)) + g(\hat{x}(t))u + Q(\hat{x}(t))^{-1}K\left[y(t) - h(\hat{x}(t))\right]$$

where $Q(x)$ is the observability matrix associated with the pair $(f(x), h(x))$:

$$Q(x) = \frac{d}{dx}\begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1}h(x) \end{bmatrix}$$

with $L_f^n$ denoting $nth$ order Lie derivative. For the linear case, *i.e.*,$h(x) = Cx$, this is equivalent to the well-known observability matrix. For simplicity, it can be assumed that the system is fully observable and states are directly measured. So that $h(x)$ becomes the identity matrix.

In robust control of nonlinear systems, disturbances are usually regarded as time dependent signals generated from an autonomous system, *i.e.*,$\dot{d}(t) = S(d)$. State estimator structure can be used to estimate this disturbance using [65]:

$$\frac{d}{dt}\begin{bmatrix} \hat{x} \\ \hat{d} \end{bmatrix} = \begin{bmatrix} f(\hat{x}(t)) + g(\hat{x}(t))\left(u + \hat{d}(t)\right) \\ S(\hat{d}(t)) \end{bmatrix} + K\left[x(t) - \hat{x}(t)\right]$$

Then using prototype functions as disturbances, such as ramps, steps, sines etc., nonlinear disturbance observers can be designed [66]. But this form is not suitable for fault detection purposes and the approach taken by Lavretsky and Hovkimyan for the adaptive control part of the ADI controller will be followed [52]. In this approach, time dependent disturbance signal is constructed using state dependent basis functions with time dependent coefficients:

$$\hat{d}(t) = W(t)^T \Phi(x(t))$$

Where $W(t)$ is an $n_b$ dimensional coefficient vector and $\Phi(x(t))$ is the set of basis functions with $n_b$ elements. Typically, Radial Basis Function (RBF) based neural network structures can be used as the basis set but actually, any integrable set of functions can be used for -at least local- approximation of the function under consideration [67].

With this structure, disturbance observer based fault detection algorithm can be formulated as follow:

$$\begin{aligned} \dot{x} &= f(x) + g(x)\left(u(t) + d(t)\right) \\ \dot{\hat{x}}(t) &= f(x(t)) + g(x(t))u(t) + \hat{d}(t) + A_s e_s(t) \\ \dot{\hat{W}}(t) &= \Gamma \text{Proj}\left(\hat{W}(t), -\Phi(x(t)) e_s^T P_s B\right) \\ \hat{d}(t) &= W(t)^T \Phi(x(t)) \\ e_s &= \hat{x}(t) - x(t) \end{aligned} \tag{2.19}$$

with $P_s$ as the solution to the Lyapunov's equation $A_s^T P_s + P_s A_s + Q = 0$ for an arbitrary $Q = Q^T > 0$. $A_s$ is the constant gain matrix that is used for the design of the disturbance estimator. $B$ is used in order to map the disturbance signal to the corresponding state.

$\Gamma$ is the adaptation constant for the estimation of the time dependent coefficients of the basis functions. Projection operator Proj is common in adaptive control structures. It is used in order to constrain the error driven $W(t)$ estimates so that they don't exceed specified bounds. Its formulation is given below [38, 68].

$$\text{Proj}\,(\theta, y) = \begin{cases} y & \text{if } f(\theta) < 0, \\ y & \text{if } f(\theta) > 0 \text{ and } \quad \nabla f^T y \leq 0, \\ y - \frac{\nabla f}{\|\nabla f\|} \langle \frac{\nabla f}{\|\nabla f\|}, y \rangle f(\theta) & \text{if } f(\theta) > 0 \text{ and } \quad \nabla f^T y > 0. \end{cases} \tag{2.20}$$

with

$$f(\theta) = \frac{(\epsilon_\theta + 1) \cdot \theta^T \theta - \theta_{max}^2}{\epsilon_\theta \cdot \theta_{max}^2}$$

$\nabla$ is the gradient operator and $\langle \cdot, \cdot \rangle$ represents the inner product. $\theta_{max}$ is the norm bound imposed on the vector $\theta$, and $\epsilon_\theta > 0$ is the projection tolerance, typically a small number such as $0.01$ is taken.

Following remarks can be made for the usage of (2.19) for fault detection (and identification) purposes:

- There are very mild requirements on the system for Luenberger-like state observers to be stable and converge to the true state. Basically, it is enough for the states to be finite and input to be bounded [64].
- Fault detection algorithm is based on online calculation of the disturbance vector $\hat{d}(t)$ and comparing with the applied control input $u(t)$. It is assumed that correlation between them indicates the presence of a fault. Since the main fault mitigation act is stopping the control inputs $u(t)$, this would also cease the disturbance signal.
- Disturbances other than faults would also lead to nonzero disturbance signals $\hat{d}(t)$, such as modeling errors or noise. These disturbance can also be compensated using adaptive parts in the fault mitigation formulation, such as:

  $$\epsilon \cdot \dot{r} = \alpha \cdot \Big[ f_1(x_1, r_2 + r) + \hat{d}(t) - f_{r1}(x_1, r_2) \Big] + A_r r$$
- RBF based neural networks are a good choice of basis functions for approximating signals with unknown structure. However for fault detection purposes, problem specific different sets might be better. One such example is provided in Chapter 3 for robotic manipulator equations.
- Projection operator Proj $(\theta, y)$ requires some conservative bounds on the time dependent coefficients. These bounds are system specific and they should be determined for the specific problem under consideration.

## 2.7 Overview of the proposed method

Theoretical analysis of the fault tolerant control structure is presented in this chapter, together with applications on analytically tractable example problems. These examples demonstrate that proposed control methodology generates smooth feedback control signals even for non-minimum phase problems. With accompanying adaptive components, proposed approach can be extended to more complex problems. These cases will be investigated in the following chapters. But it should be reminded that nonlinear systems are very diverse and system specific design steps can be taken in order to apply the presented FTC algorithm.

With this considerations, overall structure of the developed algorithmic FTC is shown in Figure 2.9 and design steps for the presented FTC structure are summarized below.

1. With appropriate FMECA analysis, determine the interconnected system form for each failure mode:

   $$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2, u_1 + d)\\ \dot{x}_2 &= f_2(x_1, x_2, u)\end{aligned}$$

   where $x_1$ corresponds to states linked to faulty actuators $u_1$ and $x_2$ corresponds to states linked to healthy actuators $u$. Depending on the system under consideration, there might be more than one valid representation of the system.

2. It should be checked that system dynamics and nominal controllers are sufficiently smooth functions of state and time and interconnected systems are controllable (Assumptions *A1* - *A3* of Theorem 2.4.1).

3. It should be verified that $\|f_1(0, x_2(t), 0)\|$ is bounded within the domain of interest. This domain should include the possible initial conditions for the fault tolerant control system. However, it should be noted that this bounded region might be exceeded during the transient response of the fault tolerant controller.

4. Construct the state estimator as given in (2.19) with appropriate gains $A_s$, $\Gamma$ and basis function set $\Phi(x_1, x_2)$.

5. Evaluate the the solution of the perturbation equation,

   $$\epsilon\dot{r} = \alpha \cdot [f_1(x_1, r_2 + r) - f_{r1}(x_1, r_2)] + A_r r$$

   and consider possible time dependent solutions $r^*(t)$. Most of the time, finding exact form of $r^*(t)$ is not necessary but it gives insight about the response of the closed loop system in time domain.

6. Calculate the system bound $\gamma_1$,

   $$\left\| \left.\frac{\partial f_1}{\partial x_2}\right|_{[x_1(t), r_2]} B \right\| \leq \gamma_1$$

7. Calculate the second system bound $\gamma_2$,

   $$\|f_1(0, x_2)\| \leq \gamma_2 \|x_2\|$$

8. Find $A_r$ that solves the Lyapunov equation $PA_r + A_r^T P + Q = 0$ with $Q$ as the identity matrix with appropriate dimensions such that

   $$\frac{1}{2\lambda_{max}(P)} > \max(\gamma_1, \gamma_2)$$

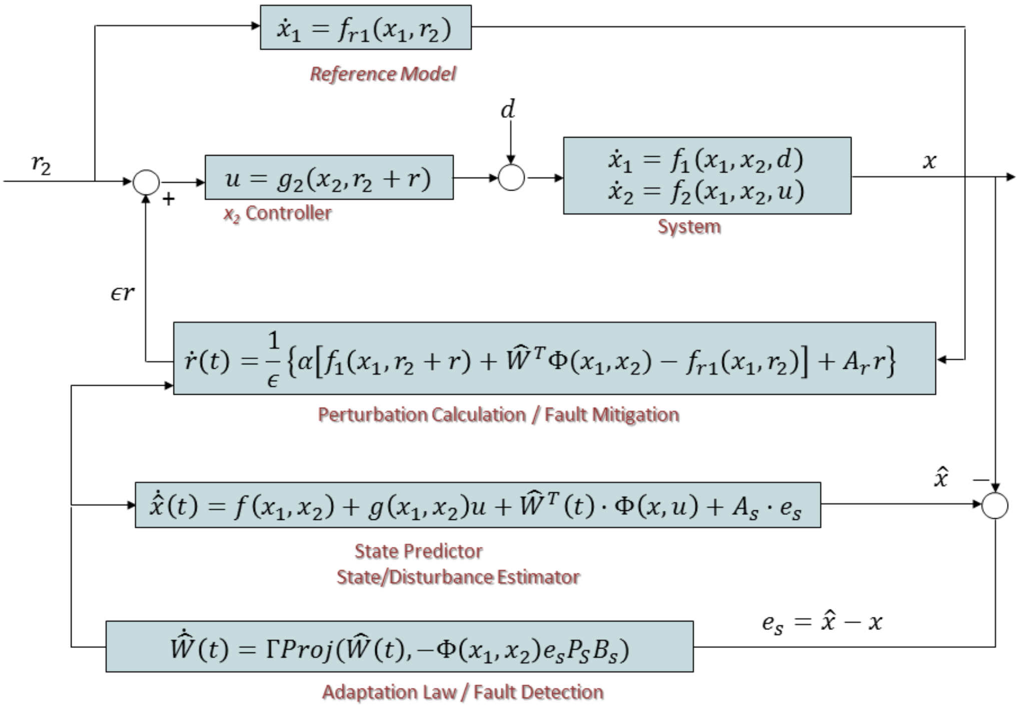

$\dot{x}_1 = f_{r1}(x_1, r_2)$
Reference Model
$d$
$r_2$
+
$u = g_2(x_2, r_2 + r)$
$x_2$ Controller
$\dot{x}_1 = f_1(x_1, x_2, d)$
$\dot{x}_2 = f_2(x_1, x_2, u)$
System
$x$
$\epsilon r$
$\dot{r}(t) = \frac{1}{\epsilon}\{\alpha[f_1(x_1, r_2 + r) + \widehat{W}^T\Phi(x_1, x_2) - f_{r1}(x_1, r_2)] + A_r r\}$
Perturbation Calculation / Fault Mitigation
$\hat{x}$
$-$
$\dot{\hat{x}}(t) = f(x_1, x_2) + g(x_1, x_2)u + \widehat{W}^T(t) \cdot \Phi(x, u) + A_s \cdot e_s$
State Predictor
State/Disturbance Estimator
$e_s = \hat{x} - x$
$\dot{\widehat{W}}(t) = \Gamma Proj(\widehat{W}(t), -\Phi(x_1, x_2)e_s P_S B_s)$
Adaptation Law / Fault Detection

Figure 2.9: Structure of the developed algorithmic fault tolerant control system

# CHAPTER 3

# FAULT TOLERANT CONTROL OF ROBOTIC MANIPULATORS

Open-chain robotic manipulators have a very diverse and wide range of application. Many industrial robots and humanoid robotic parts fall in this category. Therefore, design and control of this configuration robotic manipulators are center of focus since the first days of robotic research. Such configurations are constructed by connecting lower level joints in series with rigid links. Typically, these lower level joints are either prismatic joints that have a translational degree of freedom, or a revolute joint that have a rotational degree of freedom. With connecting one end of the chain to a static base, the motion of the other end of the manipulator, which is called “The effector end”, can be adjusted through torques and forces applied from the joints. The set of end-effector positions and orientations that can be reached with some combination of joint angles and/or positions are called the Manipulator Workspace. General overview of a robotic manipulator is shown in Figure 3.1

Industrial robotic manipulators are widely implemented especially manufacturing industry. Therefore a very rich line of study exists for fault detection and identification in robotic manipulators.

Basic elements of robotic manipulators are electrical and/or hydraulic actuators with sensors for the observation of the motion of the links. For robots working in controlled environments, like robot arms used in manufacturing processes, fault detection can be done by monitoring of external signals such as, drawn electric current, hydraulic pressure or even external video camera images. However, for robots working in unknown or changing environment, such signals may not be available or it may not be possible to discriminate faults from changing plant parameters. In order to overcome this difficulty, various techniques are developed for fault detection and

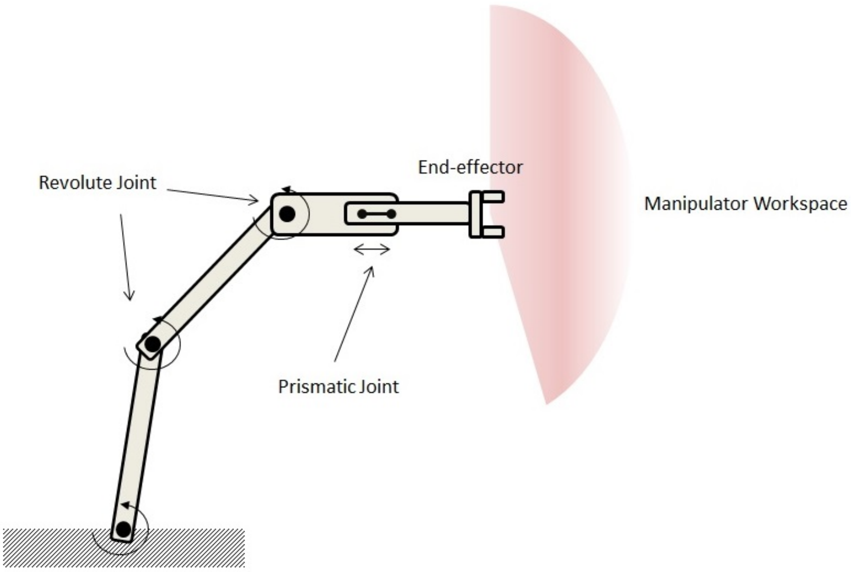


Figure 3.1: General schematic view of an open-chain robotic manipulator

identification in robotic manipulators. Early research on FDI are mostly based on analytical redundancies [69, 70], *i.e.*, using analytical relations between simultaneous sensor measurements in order to compare available sensor outputs and identification of faults from generated residual signals.

The main actuator failure mode that have been studied in the literature are related to loss of torque/force on the joints, which is called "Free swing actuator fault", uncontrolled torque/force again on the joints or locking of the joint. Free swing and uncontrolled torque failures can be modeled as an additive disturbance on the joint torques/forces as [32, 71],

$$F_i = \begin{cases} \tau_i, & \text{for free swing actuator fault} \\ \tau_i - \gamma \cdot t, & \text{for ramp actuator fault} \\ \tau_i - \tau_{max}, & \text{for saturated actuator fault} \end{cases} \tag{3.1}$$

where $\tau_i$ is the torque/force on the $\mathrm{i}^{\mathrm{th}}$ joint.

With this model, fault identification for actuator faults can be cast as an estimation problem for the external disturbance. Regression functions based on joint and velocities are used by Dixon *et al.* [71] for estimation of the fault signal. Neural networks are also widely used as functional approximator for the fault signal [72–75], due to their well known universal approximator property but alternatively Artificial Immune

System is used by Thumati *et al.* [76] and Multiple-Adaptive Neuro Fuzzy Inference System (M-ANFIS) is used by Yüksel and Sezgin [77]. Nonlinear observer structures developed specifically for robotic manipulators [32, 78] and sliding mode observers [79–82] are also available.

More recent results that are related to the present work can be summarized as follows. A terminal Sliding Mode Observer (TSMO)-based actuator fault reconstruction law is developed by Xiao and Yin [83]. As a more general approach to the problem, Mohammadi, Marquez and Tavakoli formulated a general nonlinear disturbance observer structure for Euler-Lagrange systems [84]. As an alternative structure, Oh and Chung formulated a disturbance observer structure using generalized position and generalized momentum as the state variables [85] so that disturbance signals are decoupled and a more general formulation for closed loop compensation of disturbances on rigid and flexible joint robots is developed in a later work by Kim, Park and Chung [86].

Locked joint failures put additionally, a zero velocity constraint on the states. Therefore, torque disturbance estimation schemes mentioned above, together with velocity measurements can be used for identification of locked joint failures. Apart from that Chang *et al.* posed the problem of identification of the locked joint as and optimization problem using velocity discrepancy of the end effector and used differential evolution algorithm to solve the problem [87]. Other relevant work on locked joint failures are concentrated on finding reachable subspace of the task space in the presence of locked joints [37, 88, 89] or motion planing issues on finding trajectories that are robust to joint failures [90–92].

In this part of the thesis, developed algorithmic fault tolerant control structure will be applied to the robotic manipulator case. The highlights of the structure, with advantages to the existing techniques is follows:

- A state estimator based disturbance estimator is designed. A transformation on system states is proposed, which permits decoupling of the fault signal. With this state estimator formulation, it is possible to detect and identify the actuator faults simultaneously.

- Fault mitigation algorithm is implemented as a perturbation generator on the actuated joints, in order to compensate for the motion of the faulty joints. When the internal dynamics related to the faulty states are unstable, this is a very difficult task and only configuration specific solutions exists. Developed technique provides a systematic way to design a controller even for such systems.

This chapter is organized as follows. The mathematical model of robotic manipulators are explained in Section 3.1, together with the baseline control algorithm for the normally operating system. Proposed fault detection and identification methodology is introduced in Section 3.2, while fault mitigation approach is explained in Section 3.3. Overall fault tolerant control architecture is explained and simulation results for the selected cases are presented in Section 3.5. The final remarks are drawn in Section 3.6.

## 3.1 Problem definition

### 3.1.1 Governing Equations

Newton-Euler equations are typically used for modeling of dynamic systems. But for robotic manipulators, usage of Lagrange's equation is more convenient [93]. In this formulation, equations of motion for an $m$ dimensional mechanical system can be derived from the Lagrange's equation,

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \Upsilon_i \qquad i = 1..n \tag{3.2}$$

$q \in \mathbb{R}^n$ is called the "Generalized coordinates" and $\Upsilon_i$ is the net external force acting on the $\mathrm{i}^{\mathrm{th}}$ generalized coordinate. Lagrangian function is $L(q, \dot{q}) = T(q, \dot{q}) - V(q)$, where $T$ is the kinetic energy of the system and $V$ is the potential energy of the system.

For robotic manipulators, calculation of the Lagrangian is straightforward and can be found in many text books on the subject [93, 94]:

$$L(q, \dot{q}) = \frac{1}{2} H_{ij} \dot{q}_i \dot{q}_j - V(q)$$

where, $q$ is the vector of generalized coordinates (Angle for links with rotational joints and displacement for prismatic joints). In this formulation, Einstein summation convention is used, *i.e.*, summation is implied for repeated indices.

$H_{ij}(q)$ is the inertia tensor of the manipulator and the Coriolis matrix $C_{ij}(q, \dot{q})$ of the manipulator can be formulated as,

$$C_{ij}(q, \dot{q}) = \frac{1}{2} \left( H_{ij,k} + H_{ik,j} - H_{kj,i} \right) \dot{q}_k \tag{3.3}$$

Again repeated indices imply summation over the index and derivative with respect to an index is shown with comma “,”.

With grouping the gravitational and other external forces on the joints terms under the vector $G$ and using $\tau$ as the vector of generalized forces (Torque for links with revolute joints and force for prismatic joints), governing equations for robotic manipulators can be written as,

$$H(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + G(q, \dot{q}) = \tau \tag{3.4}$$

Sometimes, more compact form of (3.4) is more suitable for manipulation of equations. For this purpose, terms can be grouped into form of $\phi(q, \dot{q}) = C(q, \dot{q}) \cdot \dot{q} + G(q, \dot{q})$.

Lagrange’s equation for robotic manipulators (3.4) have some properties that are important for control applications:

- $H(q)$ is symmetric and positive definite
- $\left( \dot{H} - 2C \right) \in \mathbb{R}^{n \times n}$ is a skew-symmetric matrix.

Second property is often referred to as the “passivity property” since it implies that in the absence of friction, the net energy of the system is conserved. These properties permit development of generalized control laws for robotic manipulators [93].

### 3.1.2 Control of robotic manipulators

#### 3.1.2.1 Computed torque technique

Although (3.4) is a coupled nonlinear dynamic system, positive definiteness of the inertia matrix $H$ permits development of rather simple control laws.

Consider a time dependent reference trajectory $(q_d, \dot{q}_d, \ddot{q}_d)$. For $e = q - q_d$, control law in the form of,

$$\tau = H(q)\left(\ddot{q}_d - K_v \dot{e} - K_p e\right) + C(q, \dot{q})\dot{q} + G(q, \dot{q}) \tag{3.5}$$

would lead to an error dynamics of,

$$H(q)\left(\ddot{e} + K_v \dot{e} + K_p e\right) = 0$$

Since $H$ is positive definite, unique solution of the above equation is $\ddot{e}+K_v\dot{e}+K_p e = 0$. Therefore tracking performance can be adjusted with specified error dynamics, with suitable choice of feedback gain matrices $K_v$ and $K_p$. Control law (3.5) is called the computed torque technique and it is a form of feedback linearization.

Skew-symmetric form of $\left(\dot{H} - 2C\right)$ permits construction of Lyapunov functions for other linear feedback laws [93]. One such configuration is the so called “Augmented PD Control” law. In this feedback control system, control force to be applied is calculated using the following formula.

$$\tau = H(q)\ddot{q}_d + C(q, \dot{q})\dot{q}_d + G(q, \dot{q}) - K_v \dot{e} - K_p e \tag{3.6}$$

#### 3.1.2.2 Partial-feedback linearization

Feedback linearization technique for control of nonlinear systems is based on the idea that if the control input is transformed into a form that cancels out the nonlinearity, then resulting system becomes linear. For example, for control affine systems, *i.e.*, when the control input is linear, such as in the form of $x = f(x) + g(x) \cdot u$, transformation

$u = g(x)^{-1}\left(v - f(x)\right)$ would lead to the linear system $\dot{x} = v$. Of course in order to do this transformation, existence of $g(x)^{-1}$ should be guaranteed.

Computed torque technique uses the same idea, if the control law (3.5) is considered with the transformation $u = Hv + C\dot{q} + G$. The resulted system is $\ddot{q} = v$. Interesting thing about robotic manipulators is that feedback linearization is possible, even when only some of the joints are actuated. In robotic research, this is usually denoted as "underactuated". This configuration is important, since it is also analogous to the condition for which some of the actuators are failed.

In order to analyze this form of the system, joints can be divided into two as actuated and unactuated. Let $q_1 \in \mathbb{R}^{n-m}$ be the set of the states of the unactuated joints and $q_2 \in \mathbb{R}^m$ be the set of the states of the actuated joints.

Using the parameter $\phi(q,\dot{q}) = C(q,\dot{q}) \cdot \dot{q} + G(q,\dot{q})$, system dynamics (3.4) can be regrouped as,

$$\begin{aligned} H_{11}(q) \cdot \ddot{q}_1 + H_{12}(q) \cdot \ddot{q}_2 + \phi_1(q,\dot{q}) =& 0 \\ H_{21}(q) \cdot \ddot{q}_1 + H_{22}(q) \cdot \ddot{q}_2 + \phi_2(q,\dot{q}) =& \tau \end{aligned} \tag{3.7}$$

In this formulation, subscripts denotes the appropriate submatrices in accordance with the grouping of the states and with suitable dimensions: $H_{11} \in \mathbb{R}^{(n-m)\times(n-m)}$, $H_{12} \in \mathbb{R}^{(n-m)\times m}$, $H_{21} \in \mathbb{R}^{m\times(n-m)}$, $H_{22} \in \mathbb{R}^{m\times m}$, $\phi_1 \in \mathbb{R}^{n-m}$, $\phi_2 \in \mathbb{R}^m$.

Since $H$ is a positive definite and symmetric matrix, any principal submatrices are also positive definite. Therefore $H_{11}$ and $H_{12}$ submatrice are also positive definite and this can be used for feedback linearization of the dynamics related to either unactuated states ($q_1$ ) or actuated states ($q_2$). This manipulation is called the "Partial Feedback Linearization" and can be formulated as follows [95].

In order to get the partial feedback linearized form for the actuated states, which is called "Collocated linearization", $\ddot{q}_1$ expression in second part of (3.7) can be cancelled out with multiplying the first equation with $H_{21}H_{11}^{-1}$ and subtracting it from the first equation. Also with using the transformation $\tau = \left(H_{22} - H_{21}H_{11}^{-1}H_{12}\right)v - H_{21}H_{11}^{-1}\phi_1 + \phi_2$, partial linearized form can be achieved:

$$\begin{aligned} \ddot{q}_1 =& - H_{11}^{-1} \cdot (H_{12} v + \phi_1) \\ \ddot{q}_2 =& v \end{aligned} \tag{3.8}$$

Feedback linearization in the unactuated states, which is called “Non-collocated linearization”, can be achieved similarly. Transformation $\tau = \left(H_{21} - H_{22} H_{12}^{+} H_{11}\right) v - H_{22} H_{12}^{+} \phi_1 + \phi_2$ leads to partial linearized form:

$$\begin{aligned} \ddot{q}_1 =& v \\ \ddot{q}_2 =& - H_{12}^{+} \cdot (H_{11} v + \phi_1) \end{aligned} \tag{3.9}$$

with superscript + denoting Moore-Penrose pseudoinverse of a matrix, *i.e.*, $A^{+} = \left(A^T A\right)^{-1} A^T$. In order Moore-Penrose pseudoinverse to be defined, $m > (n - m)$. This condition is called “Strongly Inertially Coupling” [95]. Therefore “Non-collocated linearization” is possible for only strongly inertially coupled systems, *i.e.*, when the active degrees of freedom (actuated states) is equal to or greater than passive degrees of freedom (unactuated states).

Once the partial-linearized form of the equations are formed, trajectory tracking can be achieved using a PD control law in the form of,

$$v = -K_v \dot{e} - K_p e$$

where $e = q_2 - q_{2d}$ for “collocated linearization” and $e = q_1 - q_{1d}$ for “non-collocated linearization”.

## 3.2 Fault diagnosis algorithm for robotic manipulators

### 3.2.1 Formulation

Consider the system dynamics, where failure is modeled as an additive signal,

$$H \cdot \ddot{q} + C \cdot \dot{q} + G + F = \tau \tag{3.10}$$

A state estimator for the torque disturbance $F$ can be designed from first order form of the dynamic system (3.10).

Typical approach in such situations is using the following set of variables:

$$x_1 = q_1 \qquad x_2 = q_2 \qquad x_3 = \dot{q}_1 \qquad x_4 = \dot{q}_2$$

With this set of variables, (3.10) can be written in first order as given in (3.11).

$$\dot{x} = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \mathbf{0} & -H^{-1} \cdot C \end{bmatrix} \cdot x + \begin{bmatrix} \mathbf{0} \\ -H^{-1} \cdot G \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -H^{-1} \cdot F \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ H^{-1} \cdot \tau \end{bmatrix} \tag{3.11}$$

where $\mathbf{0}$ and $\boldsymbol{I}$ are zero and identity matrices with appropriate dimensions, respectively.

The problem with the formulation (3.11) is that, additive fault is multiplied with the inverse of the inertia tensor. This transformation diffuses the fault signal $F_i$ to other channels as well, which makes the correct identification of the fault signal very difficult. Therefore an alternative formulation is developed within the scope of the thesis.

In order to isolate the fault signal $F$, following set of variables can be used:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \qquad \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = H \cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

With this formulation, time derivatives of state variables can be formulated as,

$$\begin{aligned} \dot{x}_{1,2} &= x_{2,3} \\ \dot{x}_{3,4} &= \dot{H} \cdot \dot{q} + H \cdot \ddot{q} \end{aligned} \tag{3.12}$$

In order to use the system dynamics of the robotic manipulators with (3.12), consider the following modification of the original equation (3.10):

$$H \cdot \ddot{q} + \dot{H}\dot{q} + (C - \dot{H}) \cdot \dot{q} + G + F = \tau \tag{3.13}$$

With definition of a modified version of the Coriolis Matrix $C' = C - \dot{H}$, structure of the equations can be preserved. Furthermore since $H$ depends only on $q$, $\dot{H}$ can be written in index notation as, $\dot{H}_{ij} = H_{ij,k} \cdot \dot{q}_k$. Comparing this with the definition of Coriolis matrix $C_{ij}$ in (3.3), modified version can be formulated as,

$$C'_{ij}(q, \dot{q}) = \frac{1}{2} \left( -H_{ij,k} + H_{ik,j} - H_{kj,i} \right) \dot{q}_k \tag{3.14}$$

and the first order form of the robotic manipulator equations for the state variables (3.12) can be written as,

$$\begin{aligned} \dot{x}_{1,2} &= x_{2,3} \\ \dot{x}_{3,4} &= -C' \cdot \dot{q} - G - F + \tau \end{aligned}$$

or

$$\dot{x} = \begin{bmatrix} \mathbf{0} & H^{-1} \\ \mathbf{0} & -C' \cdot H^{-1} \end{bmatrix} \cdot x + \begin{bmatrix} \mathbf{0} \\ -G \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -F \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \tau \end{bmatrix} \tag{3.15}$$

This equation is the disturbance decoupled form of the Lagrange's equation and it is more suitable for observer design, since the fault signal appears without any mapping.

Luenberger type observer can be implemented, as described in Section 2.6, using the dynamics given in (3.15) as,

$$\dot{\hat{x}} = \begin{bmatrix} \mathbf{0} & H^{-1} \\ \mathbf{0} & -C' \cdot H^{-1} \end{bmatrix} \cdot \hat{x} + \begin{bmatrix} \mathbf{0} \\ -G \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\widehat{F} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \tau \end{bmatrix} + A_s \cdot e_s \tag{3.16}$$

with observer error $e_s = \hat{x} - x$ and gain matrix $A_s$. For the estimation of fault, $\hat{D} = \begin{bmatrix} \mathbf{0} & \hat{F} \end{bmatrix}^T$ can be approximated with projection on a basis $\psi_{1..N}(\hat{x})$ as

$$\hat{D}_i \approx \sum_{j=1}^{N} W_{ij} \cdot \psi_j(\hat{x}) \tag{3.17}$$

where $N$ is the number of basis functions and subscript $i$ indicate the element of the vector. The weights of the basis functions can be updated with the error dynamics as,

$$\dot{\hat{W}} = \Gamma \cdot \text{Proj}\left(\hat{W}, -\psi(\hat{x}) \cdot e_s^T \cdot P_s\right) \tag{3.18}$$

with $P_s = P_s > 0$ solves the Lyapunov equation $A_s^T \cdot P_s + P_s \cdot A_s = -Q$ for arbitrary $Q > 0$. Proj is the projection operator, commonly used in adaptive controllers in order to bound the estimated gains. The formulation of the operator is given in Section 2.6 as (2.20).

The advantage of the proposed state observer structure (3.16) with respect to (3.11) is that fault term appears as $D = F$, instead of $D = H^{-1}F$ and therefore nonzero fault value in a specific channel ($F_i$) is not mapped to the other channel through $H^{-1}$. Furthermore, since it is in first order form, estimated fault vector $F$ also involves disturbance in velocity channel and therefore zero velocity can also be detected. This information can be used for identification of locked-joint failures, together with free swing, ramp and saturated actuated faults. Of course seeking exact matching of estimated disturbance vector $\hat{D}$ with fault model (3.1) is not feasible but it provides a framework for identification of faults.

Performance of the fault diagnosis algorithm depends on correct construction of the fault signal. Selection of basis functions to be used in (3.17) is critical for the performance of the estimator. Comparison of different basis functions is discussed and a suitable set is proposed in next section.

### 3.2.2 Selection of Basis Functions

Projecting an unknown function on a basis and online calculation of the weights is very common in adaptive control techniques. Although the formulation (3.18) is valid for any set of the basis functions, particular choice determines the number of gains that will be defined and the class of functions that can be represented. For example set of polynomials with degree up to $n$ is typically used for data fitting purposes. But, fitting to higher degree of polynomials requires exponentially increasing number of samples and therefore solutions are usually highly oscillatory. This is usually denoted as "the curse of dimensionality". Or similar to the Fast Fourier Transform of time dependent signals, Fourier series can be used as the basis set. But again, in order to represent a non-periodic signal accurately, a very high number of basis functions are required.

Gaussian Radial Basis Functions (RBF) are highly popular in adaptive control [67]. In this set, each basis function is formulated as a Gaussian function,

$$\Phi_i(x) = \exp -\frac{\|x - \xi_i\|^2}{\sigma_i^2} \tag{3.19}$$

$\xi$ and $\sigma$ can be chosen appropriate to the problem for each basis function, *i.e.*, depending on the domain of the unknown function that will be represented.

The problem with RBF networks is that they are not an orthogonal set. It is known that convergence properties of orthogonal sets are better. One such alternative is using Chebyshev polynomials as the basis set [42, 96]. Chebyshev polynomial of order $n$

can be generated recursively as[1],

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \tag{3.20}$$

with $T_0(x) = 1$ and $T_1(x) = x$ .

RBFs and Chebyshev polynomials are general sets that can be used in any problem. But any compact set of functions can be used for functional approximation [67] and there might be better sets for representation of the fault signal in the state estimator (3.15).

Considering the fault model (3.1), function that will be represented through basis functions is actually a function of the applied torque $F_i = f(u_i)$. Furthermore, since the applied torque is function of the system matrices $H$, $C$ and $G$ through the computed torque control law (3.5) or augmented PD (3.6), it is logical use basis functions composed of the terms also found in these matrices. For example, for the two-link vertical robot arm case that will be discussed in Section 3.5, system matrices involve terms in the form of $\sin(q_1), \sin(q_1+q_2), \sin(q_1+q_2)\dot{q}_2, \dot{q}_2^2$ etc. Therefore, following set of basis functions can also be used for a better signal construction performance for the two-link case (see Figure 3.5 for the definition of $q_1$ and $q_2$).

$$\begin{aligned} &\psi_1 = 1 && \psi_2 = \cos q_1 && \psi_3 = \cos(q_1+q_2) \\ &\psi_4 = \sin(q_2)\,\dot{q}_1 && \psi_5 = \sin(q_2)\,\dot{q}_2 && \psi_6 = \sin(q_2)\,\dot{q}_1^2 \\ &\psi_7 = \sin(q_2)\,\dot{q}_2^2 \end{aligned} \tag{3.21}$$

### 3.2.3 Performance comparison

In other to evaluate the performance of RBF, Chebyshev polynomial based and custom set of basis functions, numerical simulation is conducted for the two link robot arm case.

Three alternative set of basis functions are used for the construction of the fault signal $F$.

---

[1] To be more specific, this set is called the “Chebyshev polynomial of the first kind”. This form is implied when referred to Chebyshev polynomials.

#### 3.2.3.1 Radial Basis Functions (RBF) Neural Network

Fault signal is constructed using an RBF network with $\sigma_{1-4} = \sqrt{3/2}$ as,

$$\Phi_i(x) = \exp\left[-\frac{2}{3}\left(\left(q_1 - \xi_i^1\right)^2 + \left(q_2 - \xi_i^2\right)^2 + \left(\dot{q}_1 - \xi_i^3\right)^2 + \left(\dot{q}_2 - \xi_i^4\right)^2\right)\right]$$

The vector of center points $\xi_i = (\xi_i^1, \xi_i^2, \xi_i^3, \xi_i^4)$ is generated from the combinations of the following set:

$$\begin{aligned} \xi^1 &= \{-2, 0, 2\} \\ \xi^2 &= \{-2, 0, 2\} \\ \xi^3 &= \{-2, 0, 2\} \\ \xi^4 &= \{-2, 0, 2\} \end{aligned}$$

which makes the total set of basis functions 81.

#### 3.2.3.2 Chebyshev Polynomial based Orthogonal Set of Basis Functions

First three Chebyshev polynomials of the first kind are used for each state as,

$$\begin{array}{llll} T_0 = 1 & & & \\ T_1 = q_1 & T_3 = q_2 & T_5 = \dot{q}_1 & T_7 = \dot{q}_2 \\ T_2 = 2q_1^2 - 1 & T_4 = 2q_2^2 - 1 & T_6 = 2\dot{q}_1^2 - 1 & T_8 = 2\dot{q}_2^2 - 1 \end{array}$$

which makes the total set of basis functions 9.

#### 3.2.3.3 Custom Basis Set

System matrices involve terms in the form of $\sin(q_1), \sin(q_1 + q_2), \sin(q_1 + q_2)\dot{q}_2, \dot{q}_2^2$ etc. Therefore, following set of basis functions can also be used for a better signal construction performance for the two-link case:

$$\begin{array}{lll} \psi_1 = 1 & \psi_2 = \cos q_1 & \psi_3 = \cos(q_1 + q_2) \\ \psi_4 = \sin(q_2)\,\dot{q}_1 & \psi_5 = \sin(q_2)\,\dot{q}_2 & \psi_6 = \sin(q_2)\,\dot{q}_1^2 \\ \psi_7 = \sin(q_2)\,\dot{q}_2^2 & & \end{array}$$

which makes the total set of basis functions 7.

Numerical simulation is conducted for free swing type fault. System model is integrated with maximum $1\,\mathrm{ms}$ time step in continuous time, while state estimator is run with $10\,\mathrm{ms}$ time step in discrete time.

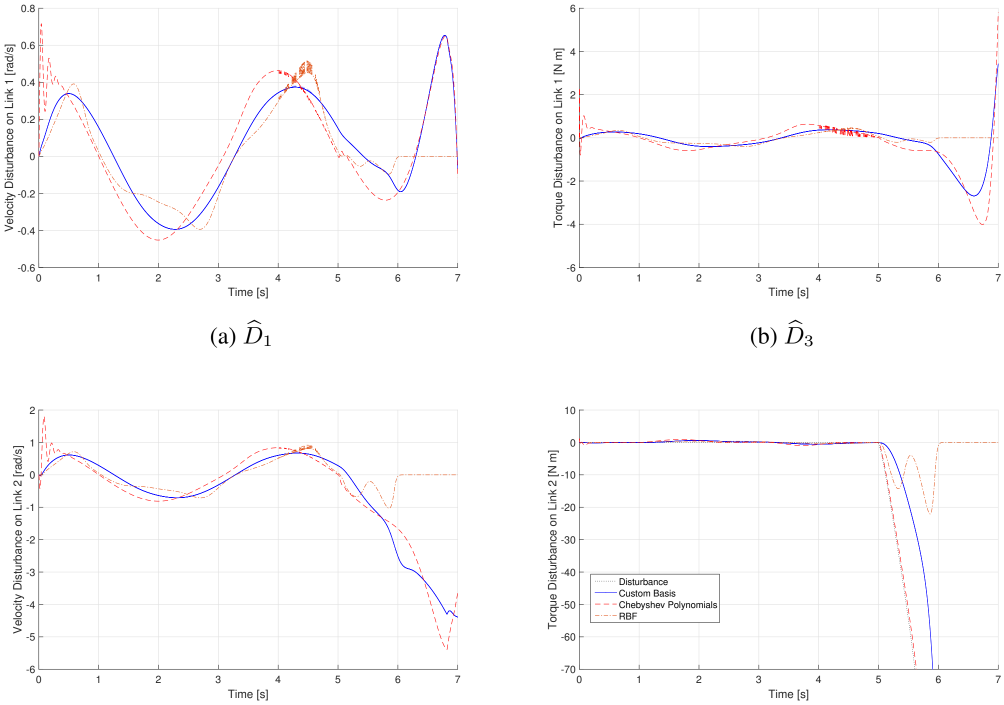
The system is started at position $(q_1, q_2) = (90°, 0)$ and sinusoidal reference commands are applied to the system such that $(r_1, r_)) = (90° + 20° \sin 2\pi t/4, -20° \sin 2\pi t/4)$. Torque loss on joint 2 case is simulated. Fault occurs at time $t = 5\,\mathrm{s}$. The results are shown in Figure 3.2.

Disturbance vector is estimated using three sets of basis functions. For the custom basis set, Chebyshev polynomials and RBF basis sets, adaptive gains are chosen as $\Gamma = 10000, 1000$ and $100000$ respectively. Projection operator is used with bound $W_{bound} = [-1000, 1000]$ and $\epsilon_\theta = 0.01$. Initial condition for the adaptive gains are taken as zero.

Fault on the second link can be observed from the disturbance torque on the second link (Figure 3.2.d). With the decoupled form of the estimator, only the fourth element is correlated with the applied torque, although some residual disturbances are present on the other elements of the disturbance vector $\widehat{D}$.

Following comments can be made on the results.

- Best performance for disturbance signal construction is achieved using Chebyshev polynomial basis set.
- For a similar convergence time, adaptive gains used for custom basis set is 10 times, RBF basis set is 100 times larger than the gain used for Chebyshev polynomial set.
- On the other hand, Chebyshev polynomial basis set has a longer initial convergence time (Approximately 1 second) and a low frequency residual signal exists on the



(a) $\widehat{D}_1$

(b) $\widehat{D}_3$

(c) $\widehat{D}_2$

(d) $\widehat{D}_4$

Figure 3.2: Estimated disturbance vector with different basis function sets for two-link vertical robot arm problem. Loss of torque occurs on the second joint at $t = 5\,\mathrm{s}$.

estimated disturbance, even during healthy operation of the manipulator.

- It seems that overall performance of the Chebyshev polynomial set is better. Fault signal can be constructed with lower adaptation gain and with fewer number of basis functions, but there is a catch. As the number of basis functions increases, order of the polynomial increases. This causes instabilities on the gain estimation and therefore adaptive gain should be lowered to compensate for that. As usual, higher performance comes with reduced robustness and therefore adaptive gain and order of the polynomial should be chosen with these considerations.

- Custom basis set is also a good alternative. Similar performance to Chebyshev polynomial basis set can be achieved without robustness problems. The situation is shown in Figure 3.3. While increasing order of the Chebyshev polynomial increases the convergence speed, stability is degraded. On the other hand, custom basis set produces smooth solutions even with very high adaptation rates. Of course using high adaptation rates has its own problems when noise is present.

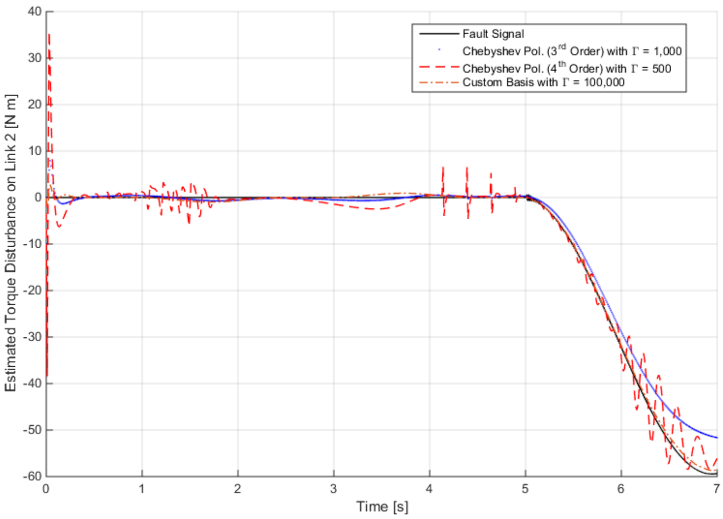


Figure 3.3: Comparison of disturbance estimation performances for the basis sets with Chebyshev polynomials up to order 3, up to order 4 and the custom basis set.

## 3.3 Design of the fault mitigation algorithm

Fault mitigation algorithm for robotic manipulators can be developed with considering the faulty links as passive links. After the detection of the fault at specific joints, powers of the faulty actuators can be cut so that the faulty link becomes unactuated. Then the stability of these passive links can be maintained with the developed control structure.

As in the case of partial feedback linearization, for a manipulator with $n$ number of links with $m$ of them have healthy actuators, system states can be grouped into two as $q_1 \in \mathbb{R}^{n-m}$ and $q_2 \in \mathbb{R}^m$ with the dynamics given in (3.7).

Consider the case where a PD controller is used for each link in the nominal controller of the manipulator. Once the faulty actuators are inactivated, the resultant system can be analyzed using the collocated partially-linearized form of the robotic equations. This equation is repeated below for convenience.

$$\begin{aligned} \ddot{q}_1 =& -H_{11}^{-1} \cdot (H_{12} v + \phi_1) \\ \ddot{q}_2 =& v \end{aligned} \tag{3.8}$$

with $v = -K_v (\dot{q}_2 - \dot{q}_{2d}) - K_p (q_2 - q_{2d})$. Since $(q_{2d}, \dot{q}_{2d})$ defines the reference dynamics for $q_2$, it is equivalent to $v = -K_v \dot{e} - K_p e$.

Comparing with the general form of the cascade system $\dot{x}_1 = f_1(x_1, x_2)$ with $x_1 = (q_1, \dot{q}_1)$, $x_1$ dynamics can be written as,

$$\begin{aligned} \begin{bmatrix} \dot{q}_1 \\ \ddot{q}_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -H_{11}^{-1}C_{11} \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ -H_{11}^{-1}\left(G_1 + C_{12}\dot{q}_2\right) \end{bmatrix} \\ + H_{11}^{-1}H_{12} \begin{bmatrix} 0 & 0 \\ K_p & K_v \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \end{aligned} \tag{3.22}$$

This is a second order system and therefore it has the dimension $2\,(n-m)$. In order compensate for them, $(n-m)$ number of healthy actuators can be chosen and perturbations dynamics can also be written as a second order system. Therefore with comparison to (2.6) in Section 2.4, this relation can be written as,

$$\begin{aligned} \epsilon \begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = \alpha \Bigg\{ \left( \begin{bmatrix} 0 & 1 \\ 0 & -H_{11}^{-1}C_{11} \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ -H_{11}^{-1}\left(G_1 + C_{12}\dot{q}_2\right) \end{bmatrix} \right)_{\substack{(q_2,\dot{q}_2)= \\ (q_{2d}+r,\dot{q}_{2d}+\dot{r})}} \\ - \left( \begin{bmatrix} 0 & 1 \\ A_{rm21} & A_{rm22} \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ B_{rm21} \end{bmatrix} r_1 \right) \Bigg\} + A_r \begin{bmatrix} r \\ \dot{r} \end{bmatrix} \end{aligned} \tag{3.23}$$

Where $r \in \mathbb{R}^{n-m}$ is the perturbation on the joint positions and $\dot{r} \in \mathbb{R}^{n-m}$ is on the joint velocities. The fault mitigation algorithm moves the system to a position where the dynamics of the passive links can also be stabilized. Therefore the reference velocity of the healthy actuators $q_{2d}$ can be taken as $0$.

The reference model for the $q_1$ dynamics is written as,

$$\begin{bmatrix} 0 & 1 \\ A_{rm21} & A_{rm22} \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ B_{rm21} \end{bmatrix} r_1$$

with $r_1$ as the final position of the faulty joints, *i.e.*, $q_1$. For a single faulty joint, $A_{rm}$ and $B_{rm}$ can be written as a second order system as,

$$A_{rm} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \qquad B_{rm} = \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix}$$

The analysis of the perturbation system (3.23) can be done from two aspects. Firstly, the time dependent solution of $(r, \dot{r})$ for $\epsilon = 0$ will be checked. Then the conditions for the exponential stability of the boundary layer model will be analyzed.

### 3.3.1 Solution of the perturbed system

For $\epsilon = 0$, (3.23) can be written as a system of algebraic equations:

$$\alpha \left\{ \left( \begin{bmatrix} 0 & 1 \\ 0 & -H_{11}^{-1}C_{11} \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ -H_{11}^{-1}\left(G_1 + C_{12}\dot{r}\right) \end{bmatrix} \right)_{(q_2,\dot{q}_2)=(q_{2d}+r,\dot{r})} \right.$$
$$\left. - \left( \begin{bmatrix} 0 & 1 \\ A_{rm21} & A_{rm22} \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ B_{rm21} \end{bmatrix} r_1 \right) \right\} + A_r \begin{bmatrix} r^* \\ \dot{r}^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

where $r^*$ and $\dot{r}^*$ are the roots of the set of equations.

The upper part of the equation is satisfied for all $(A_{r11} + r^* + A_{r12}\dot{r}^* = 0)$. In order to enforce a solution where $\dot{r}^* = 0$, $A_r$ should be chosen such that $A_{r11} = 0$ and $A_{r12} \neq 0$.

The lower part of the system of equations can be written as,

$$\alpha \left\{ \left(-H_{11}^{-1}C_{11}\dot{q}_1 - H_{11}^{-1}G_1 + C_{12}\dot{r}\right)\Big|_{(q_{2d}+r,\dot{r})} - A_{rm21}q_1 - A_{rm22}\dot{q}_1 - B_{rm21}r_1 \right\}$$
$$+A_{r21}r^* + A_{r22}\dot{r}^* = 0$$

Again consider the static solution where $\dot{r}^* = 0$. Coriolis terms $C_{11}$ and $C_{12}$ are zero. Furthermore, terms related to the reference model is zero with $q_1 = r_1$. Therefore the remaining terms are,

$$-\alpha H_{11}^{-1}G_1 + A_{r21}r^* = 0$$

$G_1$ is the gravitational moments on the first link. If the resting position of the first link $r_1$ is chosen such that $G_1|_{q_1=r_1} = 0$, then the trajectory perturbation converges to $r^* = 0$. For example, for the vertical two-link robotic manipulator, which will be analyzed in Section 3.5, upward (or downward) position of the manipulator with two links stretched on a straight line is such a configuration.

To sum up, the analysis of the reduced problem reveals that the shaping matrix $A_r$ should be chosen such that upper-left principal element is zero and the resting position of the unactuated links $(r_1)$ should be chosen such that the gravitational moments on this link is zero.

### 3.3.2 Analysis of the boundary layer problem

Robotic manipulator equations are highly coupled and in general and in order to analyze the stability of the boundary layer equations, they should be linearized. As derived in Section 2.4 in Chapter 2, linearized form of the boundary layer equation can be written as,

$$\frac{dy}{d\tau} = \left[ A_r + \alpha \left. \frac{\partial f_1}{\partial x_2} \right|_{[x_1(t), r_2]} \right] y$$

For the general robotic manipulator problem, this is a matrix equation the linearization should be done over the vector $x_2 = (q_2, \dot{q}_2)$. The derivation is rather lengthy and therefore it is provided in Appendix B. The resultant Jacobian of $f_1$ is,

$$\left. \frac{\partial f_1}{\partial x_2} \right|_{x_2=(r_2,0)} = \begin{bmatrix} 0 & 0 \\ \left. \left( H_{11}^{-1} \frac{\partial H_{11}}{\partial x_2^1} H_{11}^{-1} G_1 - H_{11}^{-1} \frac{\partial G_1}{\partial x_2^1} \right) \right|_{q_2=r_2} & 0 \end{bmatrix} \tag{3.24}$$

With this expression, boundary layer equation for robotic manipulators can be written,

$$\frac{dy}{d\tau} = \left( A_r + \alpha \begin{bmatrix} 0 & 0 \\ \left. \left( H_{11}^{-1} \frac{\partial H_{11}}{\partial q_2} H_{11}^{-1} G_1 - H_{11}^{-1} \frac{\partial G_1}{\partial q_2} \right) \right|_{q_2=r_2} & 0 \end{bmatrix} \right) y \tag{3.25}$$

$\dot{r}_2 = 0$ so that $C_{11} = C_{12} = 0$ is also used in finding the above result.

Boundary layer equation (3.25) of the perturbation problem (3.22) and (3.23) highly depends on the specific problem and they should be analyzed separately. But following comments can be made regarding the assumptions and conditions of the main theorem on the fault mitigation control structure Theorem 2.4.1.

- Robotic manipulators usually have continuous parameter sets. Therefore assumption on Lipschitz continuity of the functions, Assumption *A1* of Theorem 2.4.1 are valid for most of the problems.

- Assumption *A2* of Theorem 2.4.1 is related to the controllability of $x_1$ dynamics with $x_2$ regarded as control input. This property is system and configuration specific and therefore it is should be verified on specific application.

- Assumption *A3* of Theorem 2.4.1 is related to the controllability of $x_2$ dynamics. This assumption is valid for robotic manipulators, since this part of the system is fully actuated.
- Assumption *A4* of Theorem 2.4.1 is related to the bound on $\|f_1(0, x_2)\|$. For $x_1$ dynamics given in (3.22), $f_1(0, x_2)$ can be written as,

  $$f_1(0, x_2) = \left.\begin{bmatrix} 0 \\ -H_{11}^{-1}\left(G_1 + C_{12}\dot{q}_2\right) \end{bmatrix}\right|_{(q_1, \dot{q}_1)=(0,0)}$$

  This equation should be checked for the bound.
- Condition *C1* of Theorem 2.4.1 is related to the bound on $\left\|\frac{\partial f_1}{\partial x_2}\right\|$. This Jacobian is given in (3.24).
- For robotic manipulators without external force, such as horizontally positioned planar manipulators, $G_1 = 0$ and hence the boundary layer equation becomes $y' = A_r y$. The results is consistent with the fact that such configurations are neutrally stable.
- Determinant of the Jacobian matrix $\frac{\partial f_1}{\partial x_2}$ is zero. This poses a problem on the calculation of the norms. Usage of Frobenius norm is more appropriate in these situations. Frobenius norm can be calculated with the following formulation:

  $$\|A\|_F = \sqrt{tr\left(AA^T\right)}$$

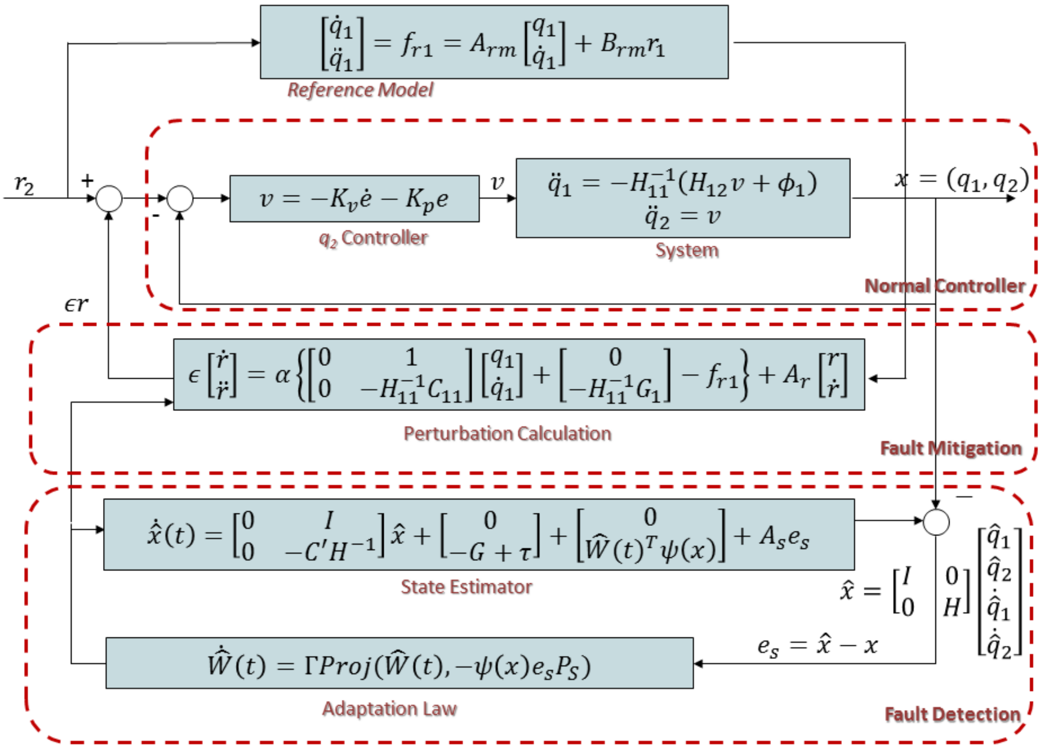
## 3.4 Fault tolerant control architecture for robotic manipulators

With the theoretical analysis presented in previous sections, application of proposed fault mitigation algorithm on robotic manipulator systems can be formulated as,

- Detection of the fault joints
- Cutting off power of the fault joints, so that they become unactuated, free-swing passive links
- Move the active links to a suitable position, where the dynamics of the passive links can also be stabilized

- Stabilize the overall system at this final position

This implementation of the above strategy is shown in Figure 3.4. Applications on selected problems are given in next section.



$\begin{bmatrix}\dot q_1\\ \ddot q_1\end{bmatrix} = f_{r1} = A_{rm}\begin{bmatrix}q_1\\ \dot q_1\end{bmatrix} + B_{rm}r_1$
Reference Model
$r_2$
+
-
$v = -K_v\dot e - K_p e$
$q_2$ Controller
$v$
$\ddot q_1 = -H_{11}^{-1}(H_{12}v + \phi_1)$
$\ddot q_2 = v$
System
$x = (q_1, q_2)$
Normal Controller
$\epsilon r$
$\epsilon\begin{bmatrix}\dot r\\ \ddot r\end{bmatrix} = \alpha\left\{\begin{bmatrix}0 & 1\\ 0 & -H_{11}^{-1}C_{11}\end{bmatrix}\begin{bmatrix}q_1\\ \dot q_1\end{bmatrix} + \begin{bmatrix}0\\ -H_{11}^{-1}G_1\end{bmatrix} - f_{r1}\right\} + A_r\begin{bmatrix}r\\ \dot r\end{bmatrix}$
Perturbation Calculation
Fault Mitigation
$\dot{\hat x}(t) = \begin{bmatrix}0 & I\\ 0 & -C'H^{-1}\end{bmatrix}\hat x + \begin{bmatrix}0\\ -G+\tau\end{bmatrix} + \begin{bmatrix}0\\ \widehat W(t)^T\psi(x)\end{bmatrix} + A_s e_s$
State Estimator
$-$
$\hat x = \begin{bmatrix}I & 0\\ 0 & H\end{bmatrix}\begin{bmatrix}\hat q_1\\ \hat q_2\\ \dot{\hat q}_1\\ \dot{\hat q}_2\end{bmatrix}$
$e_s = \hat x - x$
$\dot{\widehat W}(t) = \Gamma Proj(\widehat W(t), -\psi(x)e_sP_S)$
Adaptation Law
Fault Detection

Figure 3.4: Fault tolerant control architecture for robotic manipulators

Design steps for this system can be summarized as follows:

- For each possible fault condition, group the states that are linked to the faulty actuators as $q_1$ and the healthy ones as $q_2$.
- Construct the collocated partially linearized form of the governing equations as,

$$\begin{aligned}\ddot q_1 &= -H_{11}^{-1}\cdot(H_{12}v + \phi_1)\\ \ddot q_2 &= v\end{aligned}$$

  with auxiliary control input $v$ is calculated as,

$$v = -K_v\dot e - K_p e$$

  and the control torque vector $\tau$ is calculated as,

$$\tau = \left(H_{22} - H_{21}H_{11}^{-1}H_{12}\right)v - H_{21}H_{11}^{-1}\phi_1 + \phi_2$$

- Formulate the Luenberger type state estimator,

$$\dot{\hat{x}} = \begin{bmatrix} \mathbf{0} & H^{-1} \\ \mathbf{0} & -C' \cdot H^{-1} \end{bmatrix} \cdot \hat{x} + \begin{bmatrix} \mathbf{0} \\ -G \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\hat{F} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \tau \end{bmatrix} + A_s \cdot e_s$$

  where,

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \qquad \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = H \cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

  and the observer error is $e_s = \hat{x} - x$. The observer gain matrix $A_s$ can be chosen as a Hurwitz matrix.

- Choose appropriate set of basis functions $\Psi_{1..N}(x)$ for the construction of the fault signal as

$$\hat{F}_i = \sum_{j=1}^{N} W_{ij} \cdot \psi_j(\hat{x})$$

  Weights can be estimated using the differential equation,

$$\dot{\hat{W}} = \Gamma \cdot \text{Proj}\left(\hat{W}, -\psi(\hat{x}) \cdot e_s^T \cdot P_s\right)$$

  with $P_s = P_s > 0$ solves the Lyapunov equation $A_s^T P_s + P_s A_s = -Q$ for $Q = \boldsymbol{I}$.

- Once a fault is detected, corresponding faulty link is passivized so that the partial-linearized form of the dynamics is valid. The perturbation signal on the actuated links can be calculated as,

$$\begin{aligned} \epsilon \begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = \alpha \Bigg\{ & \left( \begin{bmatrix} 0 & 1 \\ 0 & -H_{11}^{-1} C_{11} \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ -H_{11}^{-1} G_1 \end{bmatrix} \right)_{(q_2, \dot{q}_2) = (r_2 + r, 0)} \\ & - \left( A_{rm} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + B_{rm} r_1 \right) \Bigg\} + A_r \begin{bmatrix} r \\ \dot{r} \end{bmatrix} \end{aligned}$$

  $r_1$ and $r_2$ are the resting position of the links. If possible, they should be chosen such that the gravitational effects becomes zero, *i.e.*, $G_1(r_1, r_2) = 0$.

  $A_r$ should be chosen such that upper-left principal element is zero.

- For the stability of the boundary layer equation (Conditions *C1* and *C2* of Theorem 2.4.1), evaluate the following terms:

$$\left.\frac{\partial f_1}{\partial x_2}\right|_{x_2 = (r_2, 0)} = \begin{bmatrix} 0 & 0 \\ \left.\left( H_{11}^{-1} \frac{\partial H_{11}}{\partial q_2} H_{11}^{-1} G_1 - H_{11}^{-1} \frac{\partial G_1}{\partial q_2} \right)\right|_{q_2 = r_2} & 0 \end{bmatrix}$$

$$f_1(0, x_2) = \left. \begin{bmatrix} 0 \\ -H_{11}^{-1}\left(G_1 + C_{12}\dot{q}_2\right) \end{bmatrix} \right|_{(q_1,\dot{q}_1)=(0,0)}$$

Choose $A_r$ such that, $A_r$ solves the Lyapunov equation $PA_r + A_r^T P + Q = 0$ with $Q$ as the identity matrix with appropriate dimensions that satisfies,

$$\left\| \left. \frac{\partial f_1}{\partial x_2} \right|_{x_2=(r_2,0)} \right\|_F \leq \gamma_1 \qquad \left\| f_1(0, x_2) \right\| \leq \gamma_2 \left\| x_2 \right\|$$

$$\frac{1}{2\left\| P \right\|_F} > \max(\gamma_1, \gamma_2)$$

Frobenius norm should be used in the calculations, which can be formulated as,

$$\left\| A \right\|_F = \sqrt{tr\left(AA^T\right)}$$

## 3.5 Application on selected problems

Two problems are chosen in order to demonstrate the applications of the proposed fault mitigation strategy. The first problem is the vertical planar two-link robotic manipulator problem, which is extensively studied in the literature, especially within the first years of the robot control research [97]. This is a basic configuration yet poses important characteristics of manipulator dynamics with external gravitational forces. The second configuration is the horizontally placed three-link planar manipulator. This is also an elementary configuration and it is called SCARA manipulator. Application on higher dimensional cases are demonstrated with this configuration.

Design steps and theoretical analyses are provided in the following paragraphs, together with simulation results.

### 3.5.1 Vertical two-link robot arm

Two-link robot arm is a simple configuration; two rotary actuators manipulate the motion of two rigid arms (Figure 3.5). Furthermore, many interesting robotic systems can be formulated analogues to the two-link robot arm problem such as model for legged motion, human posture while standing, cart-pole system (segway) [95]. In another interesting application, a bird perched on a branch is modeled as a two-link pendulum and its control algorithm is developed using $\mathcal{L}_1$ adaptive control technique [98]. Therefore it is an elementary configuration that is suitable as a prototype problem for development of robotic control algorithms (*e.g.*,following is a small set of articles that are published within the last five years: [99–104]).

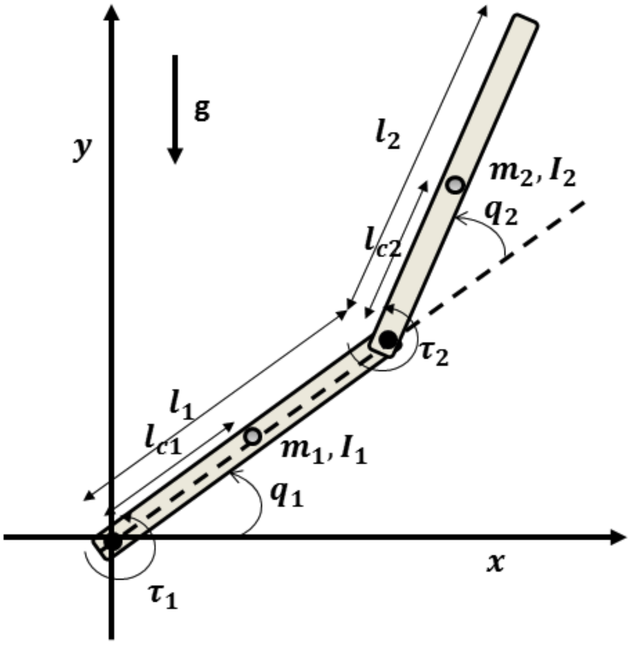


Figure 3.5: Vertical two-link robot manipulator system

Locked joint failures basically converts the system to a single link with different inertial properties and it is not very interesting from theoretical point of view. Therefore free-swing type faults are addressed in this section.

Free-swing type fault occurs when an actuator loses power and the associated link becomes a passive one and overall system becomes an underactuated system. For the two-link vertical arm problem, faulty link is unstable for the inverted position ($180^o > q_1 > 0^o$ as shown in Figure 3.5). Even though the position of the healthy link can be adjusted for a suitable position, still the unactuated link would move under the effect of gravity. Therefore, stabilizing the motion of the two links using only a single actuator is a very interesting problem and it is widely studied in the literature. Such cases are often denoted as gymnast robots, whose name derived from the resemblance of the motion of the robot to the movements of a gymnast. For the two-link case, when the actuation is from the shoulder, the system is called “Pendubot” and it is “Acrobot” when the actuation is from elbow.

Most studied problem in the underactuated two-link robot manipulator control is moving the system from downward stable position to upward unstable position and maintaining the posture in this position. In one of the first works on the subject, Spong used energy pump-up strategy through swinging of the actuated arm in Acrobot problem and LQR based linear controller in order to stabilize the system around the upward unstable equilibrium point [97]. The most comprehensive result for this problem is achieved by Lai *et al.* , where two Lyapunov functions are used for the inversion of an acrobot and drive to the system in the vicinity of the inverted straight position [105]. Stabilization around the upward position is still achieved using linearized controllers.

#### 3.5.1.1 Problem formulation

Lagrangian form for the robotic manipulators is valid for this problem. It is repeated below for convenience.

$$H(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + G(q, \dot{q}) = \tau \tag{3.4}$$

For the vertical two-link manipulator case, elements of the system matrices $H$, $C$, $G$ and $\tau$ are given below with the parameter definitions shown in Figure 3.5 [15].

$$\begin{aligned}
H_{11} &= I_1 + I_2 + m_1 \cdot l_{c1}^2 \\
&\quad + m_2 \cdot \left(l_1^2 + l_{c2}^2 + 2 \cdot l_1 \cdot l_{c2} \cdot \cos(q_2)\right) \\
H_{12} &= H_{21} = I_2 + m_2 \cdot \left[l_{c2}^2 + l_1 \cdot l_{c2} \cdot \cos(q_2)\right] \\
H_{22} &= I_2 + m_2 \cdot l_{c2}^2 \\
G_1 &= (m_1 \cdot l_{c1} + m_2 \cdot l_1) \cdot g \cdot \cos(q_1) \\
&\quad + m_2 \cdot l_{c2} \cdot g \cdot \cos(q_1 + q_2) \\
G_2 &= m_2 \cdot l_{c2} \cdot g \cdot \cos(q_1 + q_2) \\
\tau &= [\tau_1 \quad \tau_2]^T
\end{aligned}$$

Also

$$C = \begin{bmatrix} -h\dot{q}_2 & -h\left(\dot{q}_1 + \dot{q}_2\right) \\ h\dot{q}_1 & 0 \end{bmatrix} \qquad C' = \begin{bmatrix} -3h\dot{q}_2 & -h\left(\dot{q}_1 + 2\dot{q}_2\right) \\ h\left(\dot{q}_1 + \dot{q}_2\right) & 0 \end{bmatrix}$$

with

$$h = m_2 l_1 l_{c2} \sin q_2$$

where subscripts on $H$, $C$ and $G$ denotes the submatrices and $m$, $I$, $l$, $l_c$ denotes mass and inertia of the link, link length and length of the center of gravity from the pivot point, respectively.

#### 3.5.1.2 Design of the fault detection algorithm

Fault detection algorithm is based on a state estimator formulation. As explained in Section 3.2, the states used in the estimator are redefined in order to avoid coupling of the fault signal:

$$x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & H_{11} & H_{12} \\ 0 & 0 & H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

The state estimator can be formulated as,

$$\dot{\hat{x}} = \begin{bmatrix} \mathbf{0} & H^{-1} \\ \mathbf{0} & -C'H^{-1} \end{bmatrix} \cdot \hat{x} + \begin{bmatrix} \mathbf{0} \\ -G + \tau \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\hat{W}(t)^T \psi(x) \end{bmatrix} + A_s e_s \tag{3.26}$$

with the adaptation law,

$$\dot{\hat{W}} = \Gamma \cdot \text{Proj}\left(\hat{W}, -\psi(\hat{x}) \cdot e_s^T \cdot P_s\right)$$

where $P_s = P_s > 0$ solves the Lyapunov equation $A_s^T P_s + P_s A_s = -Q$ for $Q = \boldsymbol{I}$.

First four Chebyshev polynomials of first kind are used as the basis set:

$$\begin{array}{llll} \psi_0 = 1 & & & \\ \psi_1 = q_1 & \psi_4 = q_2 & \psi_7 = \dot{q}_1 & \psi_{10} = \dot{q}_2 \\ \psi_2 = 2q_1^2 - 1 & \psi_5 = 2q_2^2 - 1 & \psi_8 = 2\dot{q}_1^2 - 1 & \psi_{11} = 2\dot{q}_2^2 - 1 \\ \psi_3 = 4q_1^3 - 3q_1 & \psi_6 = 4q_2^3 - 3q_2 & \psi_9 = 4\dot{q}_1^3 - 3\dot{q}_1 & \psi_{12} = 4\dot{q}_2^3 - 3\dot{q}_2 \end{array} \tag{3.27}$$

#### 3.5.1.3 Design of the fault mitigation algorithm

Design of the fault mitigation algorithm involves determination of the parameters involved in the perturbation equation (3.23). For the case under study, the system have 2 degrees of freedom. Therefore submatrices in the formulations become scalar quantities. $[A_{rm}, B_{rm}]$ becomes a second order system and therefore it can be written as,

$$A_{rm} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \qquad B_{rm} = \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix}$$

so that the reference reference $q_1$ dynamics becomes,

$$\omega_n^2\ddot{q}_1 + 2\xi\omega_n\dot{q}_1 + q_1 = r_1$$

The fault mitigation algorithm is designed for the inverted position of the system and the fault mitigation algorithm will drive the system to the upward vertical position. Therefore, the reference commands are $(r_1, r_2) = (90^o, 0)$. In accordance with the formulations given in Section 3.5.1.1, $C_{11} = 0$ in this position.

With this results, perturbation dynamics (3.23) can be written for the vertical two-link robot arm problem as,

$$\epsilon \begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = \alpha \left\{ \left( \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ -H_{11}^{-1}G_1 \end{bmatrix} \right)_{q_2=r} - \left( \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} r_1 \right)_{r_1=90^o} \right\} + A_r \begin{bmatrix} r \\ \dot{r} \end{bmatrix}$$

Selection of $\alpha$ depends on the ease of stabilization of the perturbation dynamics. Since $H_{11}$ is positive, choice of $\alpha$ among $\pm 1$ depends on the sign of $G_1(q_1, q_2 = r)$, which can be formulated as,

$$G_1(q_1, q_2 = r) = (m_1 \cdot l_{c1} + m_2 \cdot l_1)\, g \cos(q_1) + m_2 \cdot l_{c2} \cdot g \cdot \cos(q_1 + r)$$

For the operating condition around $q_1 = 90^o$, $G_1$ does not have a definite sign. It can be positive or negative. Therefore there is no preferred sign for $\alpha$. $\alpha = 1$ can be taken for convenience.

Determination of $A_r$ is related to the Conditions *C1* and *C2* of Theorem 2.4.1 with,

$$\left.\frac{\partial f_1}{\partial x_2}\right|_{x_2=(r_2,0)} = \begin{bmatrix} 0 & 0 \\ \left.\left(H_{11}^{-1}\frac{\partial H_{11}}{\partial q_2}H_{11}^{-1}G_1 - H_{11}^{-1}\frac{\partial G_1}{\partial q_2}\right)\right|_{q_2=r_2} & 0 \end{bmatrix}$$

$$f_1(0,x_2) = \left.\begin{bmatrix} 0 \\ -H_{11}^{-1}\left(G_1 + C_{12}\dot{q}_2\right) \end{bmatrix}\right|_{(q_1,\dot{q}_1)=(0,0)}$$

Expressions for the specific terms of this equation, for the two-link problem can be written as,

$$\begin{aligned}
\left.\frac{\partial H_{11}}{\partial q_2}\right|_{q_2=0} &= -2m_2l_1l_{c2}\sin\left(q_2\right)\big|_{q_2=0} = 0 \\
\left.\frac{\partial G_1}{\partial q_2}\right|_{q_2=0} &= -m_2l_{c2}g\sin\left(q_1+q_2\right)\big|_{q_2=0} = -m_2l_{c2}g\sin q_1 \\
G_1\big|_{q_2=0} &= \left(m_1l_{c1}+m_2l_1+m_2l_{c2}\right)g\cos q_1 \\
G_1\big|_{q_1=0} &= -m_2l_{c2}g\sin q_2 \\
C_{12}\big|_{q_1,\dot{q}_1=0} &= -m_2l_1l_{c2}\sin q_2\dot{q}_2
\end{aligned}$$

With these expressions, $\left\|\left.\frac{\partial f_1}{\partial x_2}\right|_{x_2=(r_2,0)}\right\|$ can be written as,

$$\begin{aligned}
\left.\frac{\partial f_1}{\partial x_2}\right|_{x_2=(r_2,0)} &= \begin{bmatrix} 0 & 0 \\ -m_2l_{c2}g\sin q_1 & 0 \end{bmatrix} \\
&\Rightarrow \left\|\left.\frac{\partial f_1}{\partial x_2}\right|_{x_2=(r_2,0)}\right\| = \left|m_2l_{c2}g\sin q_1\right| \leq m_2l_{c2}g
\end{aligned}$$

The $\|f_1(0,x_2)\|$ term can be evaluated as,

$$\begin{aligned}
f_1(0,x_2) &= \begin{bmatrix} 0 \\ -H_{11}^{-1}\left(-m_2l_{c2}g\sin q_2 - m_2l_1l_{c2}\sin q_2\dot{q}_2^2\right) \end{bmatrix} \\
&\Rightarrow \|f_1(0,x_2)\| = \left|H_{11}^{-1}m_2l_{c2}\sin q_2\left(g+l_1\dot{q}_2^2\right)\right|
\end{aligned}$$

Since $-1 \leq \sin q_2, \cos q_2, \leq 1$, $H_{11}^{-1}$ is bounded by,

$$H_{11}^{-1} \leq \frac{1}{I_1+I_2+m_1\cdot l_{c1}^2+m_2\cdot\left(l_1^2+l_{c2}^2-2\cdot l_1\cdot l_{c2}\right)} \leq 1$$

and $\|f_1(0,x_2)\|$ is bounded by,

$$\|f_1(0,x_2)\| \leq \left|\frac{m_2l_{c2}}{I_1+I_2+m_1\cdot l_{c1}^2+m_2\cdot\left(l_1^2+l_{c2}^2-2\cdot l_1\cdot l_{c2}\right)}\left(g+l_1\dot{q}_2^2\right)\right|$$

With these expressions and with the discussions provided in Section 3.3, selection criteria for $A_r$ can be summarized as,

- $A_r$ should be in the form of,
$$A_r = \begin{bmatrix} 0 & c \\ a & b \end{bmatrix}$$
- $A_r$ should be a Hurwitz matrix. Its characteristic polynomial is $\lambda^2 - b\lambda - ac$ and it can be shown using Routh-Hurwitz criteria that; $a, b < 0$ and $c > 0$.
- Choose $A_r$ such that, $A_r$ solves the Lyapunov equation $PA_r + A_r^T P + Q = 0$ with $Q$ as the identity matrix with appropriate dimensions that satisfies,
$$\left\| \left. \frac{\partial f_1}{\partial x_2} \right|_{x_2=(r_2,0)} \right\|_F \leq \gamma_1 \qquad \|f_1(0,x_2)\| \leq \gamma_2 \|x_2\|$$
$$\frac{1}{2\|P\|_F} > \max(\gamma_1, \gamma_2)$$

#### 3.5.1.4 Simulation results

Free-swing fault on first joint (Loss of torque) case is studied as an example problem. In conjunction with Figure 3.5, numerical values used in the simulations are provided in Table 3.1,

Table 3.1: Simulated two-link robot arm's parameters

| $\boldsymbol{l_1}$ | $\boldsymbol{l_{c1}}$ | $\boldsymbol{m_1}$ | $\boldsymbol{I_1}$ |
|---|---|---|---|
| 1 m | 0.5 m | 1 kg | 0.0833 kg m$^2$ |
| $\boldsymbol{l_2}$ | $\boldsymbol{l_{c2}}$ | $\boldsymbol{m_2}$ | $\boldsymbol{I_2}$ |
| 1 m | 0.5 m | 1 kg | 0.0833 kg m$^2$ |

With this numerical values, $\gamma_1 = m_2 l_{c2} g = 4.905$ and

$$\|f_1(0,x_2)\| \leq \left|0.75\left(g + \dot{q}_2^2\right)\right|$$

For $\dot{q}_2 > g$, it can be written that,

$$\|f_1(0,x_2)\| \leq \left|0.75\left(g + \dot{q}_2^2\right)\right| \leq \dot{q}_2^2$$

Therefore there is no ultimate bound $\gamma_2 \quad \forall \dot{q}_2 \in [0, \infty)$. But by choosing a suitable bounded domain in $\mathbb{R}$, $\gamma_2$ can be determined as $\gamma_2 = \sup \dot{q}_2(t)$. Of course this limit should be checked through simulations. For the case under study, $\sup \dot{q}_2(t) = 50$ is a suitable limit.

With these values, criteria for $A_r$ can be calculated as,

$$\max(\gamma_1, \gamma_2) = 50, \qquad \Rightarrow 0.02 > \|P\|_F .$$

In order to find the appropriate $A_r = \begin{bmatrix} 0 & c \\ a & b \end{bmatrix}$, solution of the Lyapunov equation is checked for different values of $a, b, c$ parameters. Contour plot of $\|P\|_F$ for $c = 10$ is shown in Figure 3.6.

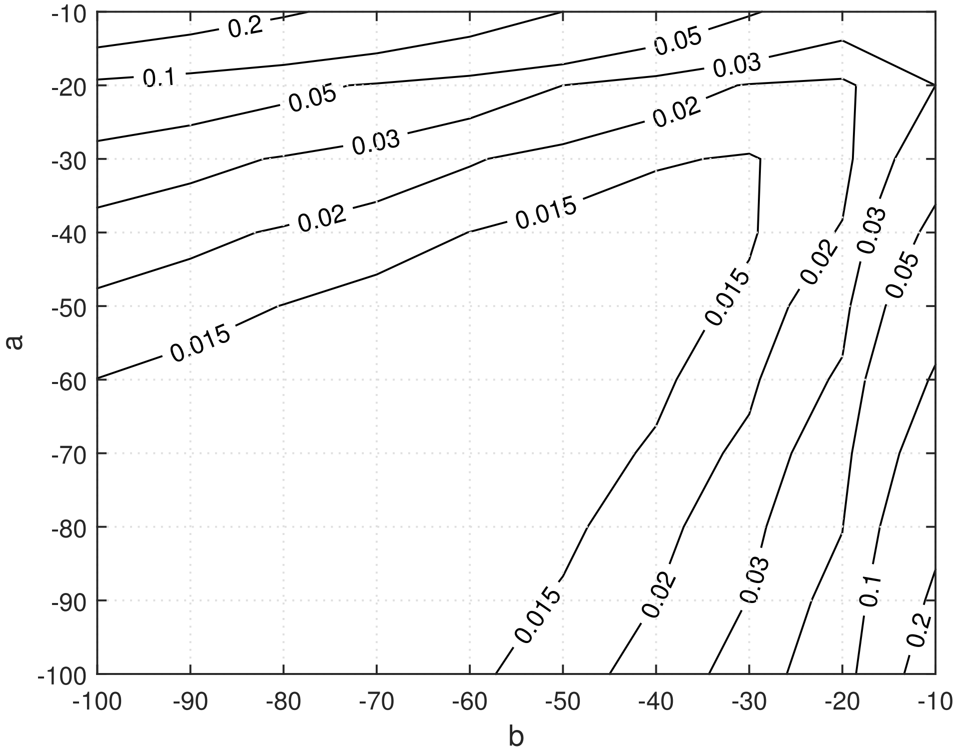


Figure 3.6: Contour plot of $\|P\|_F$ for different values of $a$ and $b$, where $P$ is the solution to the Lyapunov equation $PA_r + A_r^T P + Q = 0$ with $Q$ as the identity matrix and $A_r$ in the form of $A_r = \begin{bmatrix} 0 & c \\ a & b \end{bmatrix}$ with $c = 10$.

Computed torque based control law is used as the nominal controller with feedback gain matrices $K_v$ and $K_p$ as,

$$K_p = \begin{bmatrix} 3947.8 & 0 \\ 0 & 3947.8 \end{bmatrix} \qquad K_v = \begin{bmatrix} 125.6637 & 0 \\ 0 & 125.6637 \end{bmatrix}$$

For fault detection algorithm; first four Chebyshev polynomials of first kind, as given in (3.27), are used as the basis set for the construction of the disturbance signal. State estimator is constructed with,

$$A_s = \begin{bmatrix} -30 & 0 & 0 & 0 \\ 0 & -30 & 0 & 0 \\ 0 & 0 & -50 & 0 \\ 0 & 0 & 0 & -50 \end{bmatrix} P_s = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$

and $\Gamma_W = 1000$, $\epsilon_\theta = 0.01$. Disturbance threshold of $10\,\mathrm{N\,m}$ is used for fault detection.

For the fault mitigation algorithm, following system is used as the reference model, which corresponds to a second order system with $\xi = 0.7$ and $\omega_n = 10\,\mathrm{Hz}$.

$$A_{rm} = \begin{bmatrix} 0 & 1 \\ -3947.84 & -87.97 \end{bmatrix} \qquad B_{rm} = \begin{bmatrix} 0 \\ 3947.84 \end{bmatrix}$$

Parameters of the nonlinear controller are as follows,

$$A_r = \begin{bmatrix} 0 & 10 \\ -40 & -40 \end{bmatrix} \qquad \alpha = 1 \quad \varepsilon = 0.5$$

Partial feedback linearized control law is used for application of the control signals. Following control gains are used, which corresponds to an error dynamics with $\xi = 1.4$ and $\omega_n = 5\,\mathrm{Hz}$.

$$K_1 = 87.97 \qquad K_2 = 986.96$$

Simulation is conducted for a hypothetical case where two-link arm is started at the position $(q_1, q_2) = (90°, 0°)$ and links are moved to the position $(q_1, q_2) = (80°, 20°)$ using a nominal controller. Fault occurs at $t = 1sec.$. Simulation results are shown in Figure 3.7, Figure 3.8, Figure 3.9 and Figure 3.10 for the link positions, angular velocities, perturbation signal and the torque output of the controller respectively.

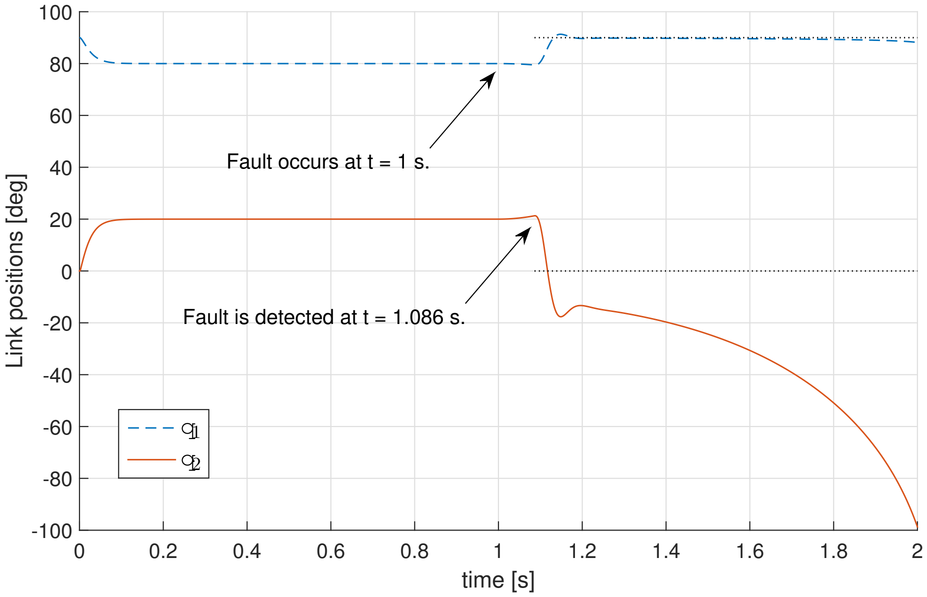


Figure 3.7: Link position for the two-link robot arm problem simulation

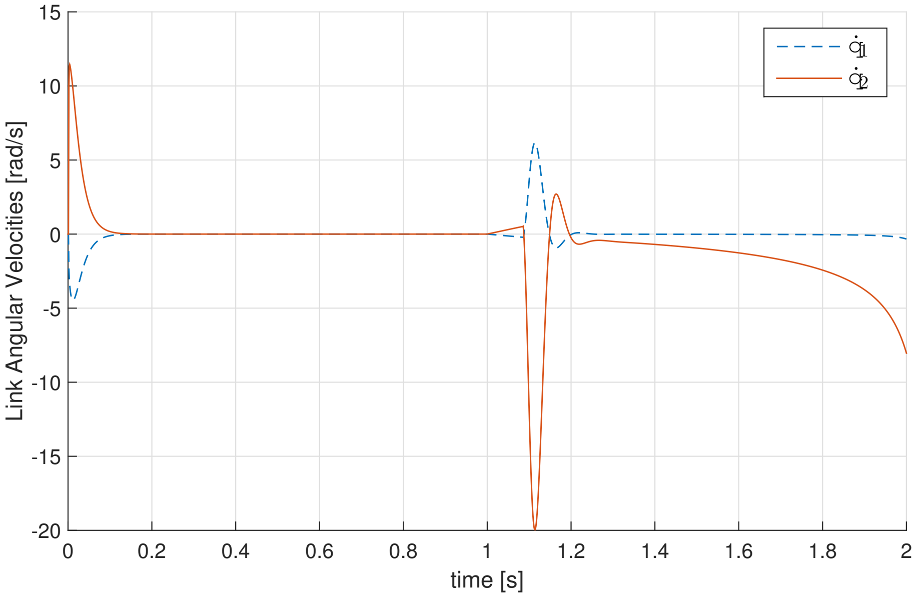


Figure 3.8: Link angular velocities for the two-link robot arm problem simulation

Fault detection algorithm detects the fault at $t = 1.08sec.$ and FTC drives the system to the upward position $(q_1, q_2) = (90°, -15°)$. Although system starts to converge to the position $(q_1, q_2) = (90°, -16.5°)$, $q_2$ dynamics diverges, although $q_1$ maintains its state a little bit longer.

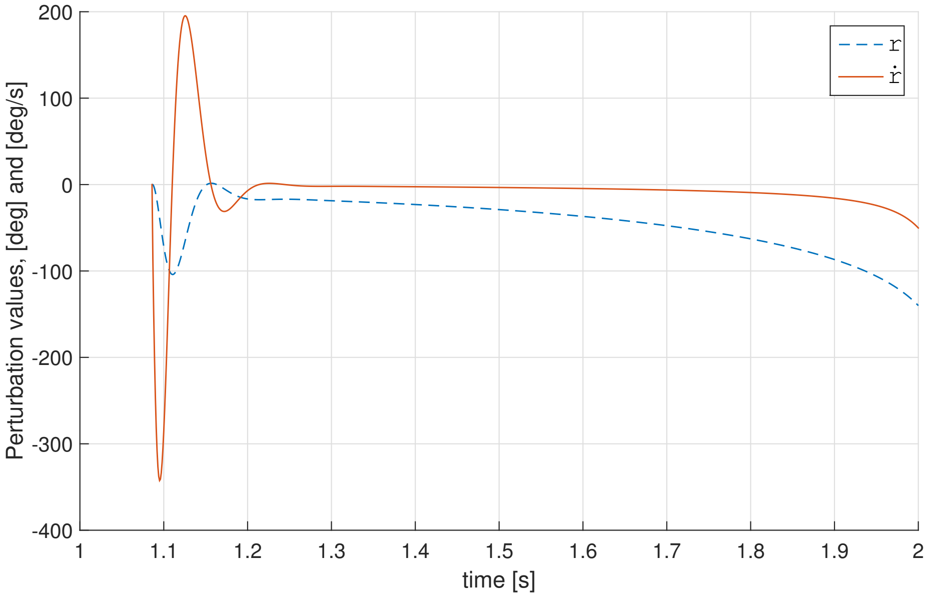


Figure 3.9: Perturbation on the second links position ($r$), for the compensation of fault on the first joint

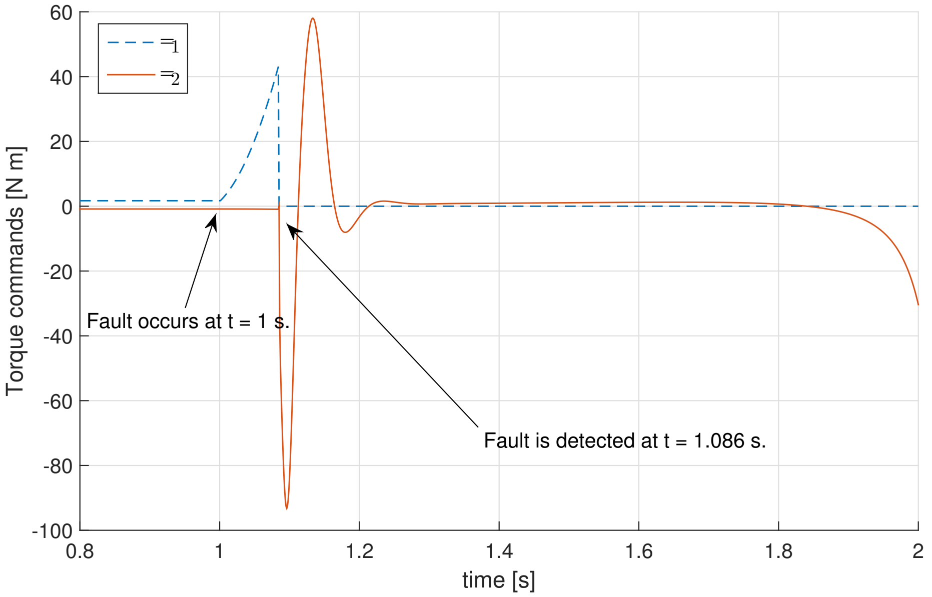


Figure 3.10: Applied torques for the two-link robot arm problem simulation

Although this result seems surprising, the situation is clearer when the free-body diagram of the system is considered (Figure 3.11).

When the system is at the upward position, *i.e.*, $q_1 = 90°$, moment caused by the

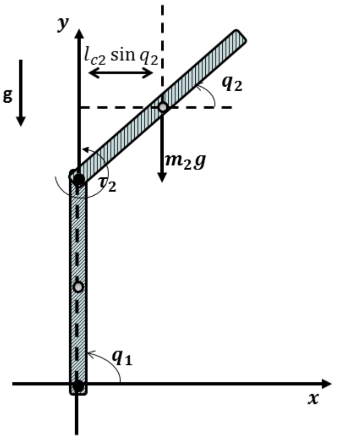


Figure 3.11: Moments on the links of the vertical two-link robot arm for the upward position

weight of the second link around joint 1 is the same as the moment around joint 2. Therefore, when the applied torque on joint 2 ($\tau_2$) counters this moment, net moment around joint 1 also becomes zero. This situation is independent of the second links position ($q_2$). This creates infinite number of equilibrium points that corresponds to different $(q_2, \tau_2, )$ duplets that satisfy the condition $\tau_2 = m_2 q l_{c2} \sin q_2$. Solution of the perturbation problem moves along these equilibrium points and finally diverge.

A favourable behaviour for the system would be moving to the other direction on the phase space towards the finally resting point of $(q_1, q_2) = (90°, 0°)$, unfortunately this is not the case with the current formulation.

A possible solution to this problem is usage of adaptive terms to compensate for this behaviour. As shown in Chapter 2, control system formulation is suitable for such kind of modifications. For this purpose, a reference model for the system is constructed as,

$$A_{rm} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_n^2 & -2\xi\omega_n & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \qquad B_{rm} = \begin{bmatrix} 0 & 0 \\ \omega_n^2 & 0 \\ 0 & \omega_n^2 \end{bmatrix}$$

such that the ideal dynamics of the system becomes,

$$\begin{bmatrix} \dot{q}_{1ref} \\ \ddot{q}_{1ref} \\ \dot{q}_{2ref} \\ \ddot{q}_{2ref} \end{bmatrix} = A_{rm} \begin{bmatrix} q_{1ref} \\ \dot{q}_{1ref} \\ q_{2ref} \\ \dot{q}_{2ref} \end{bmatrix} + B_{rm} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

As in the MRAC formulation and similar to the state estimator formulation, the error vector,

$$e = \begin{bmatrix} q_{1ref} \\ \dot{q}_{1ref} \\ q_{2ref} \\ \dot{q}_{2ref} \end{bmatrix} - \begin{bmatrix} q_1 \\ \dot{q}_1 \\ q_2 \\ \dot{q}_2 \end{bmatrix}$$

is used for calculation of weights on an appropriate basis using the update law

$$\dot{\hat{W}} = \Gamma \cdot \text{Proj}\left(\hat{W}, -\psi(q_1, q_2) \cdot e^T \cdot P\right)$$

Then the adaptive compensation term can be calculated as,

$$D_i = \sum_{j=1}^{N} W_{ij} \cdot \psi_j(q_1, q_2)$$

and the perturbation dynamics can be written as,

$$\begin{aligned} \epsilon \begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} &= \alpha \Bigg\{ \left( \begin{bmatrix} 0 & 1 \\ 0 & -H_{11}^{-1} C_{11} \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ -H_{11}^{-1} G_1 \end{bmatrix} \right)_{(q_2, \dot{q}_2) = (r_2 + r, 0)} \\ &\quad - \left( A_{rm} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + B_{rm} r_1 \right) + D \Bigg\} + A_r \begin{bmatrix} r \\ \dot{r} \end{bmatrix} \end{aligned}$$

The result achieved with this control system is shown in Figure 3.12. Although the adaptive term considerable improves the system response and drives the states to the desired $(q_1, q_2) = (90°, 0°)$ position, controller cannot maintain this position. Deviation of first link ($q_1$) from the ideal upward position again causes divergence of the overall system.

As a final remark, upward position is an ill conditioned situation and it should be mentioned that existing controllers in the literature also switches to linearized controllers at this part of the state-space.

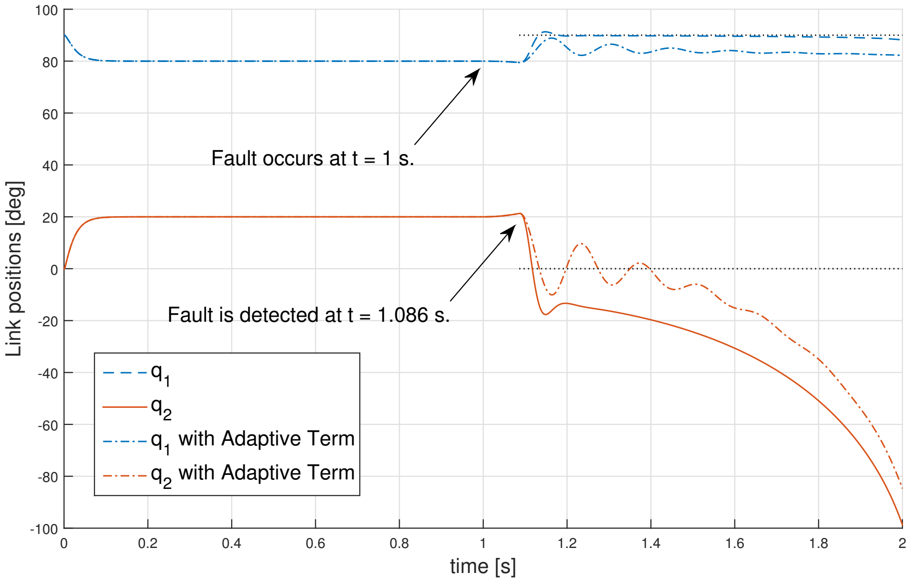


Figure 3.12: Link position for the two-link robot arm problem simulation

### 3.5.2 Horizontal three-link robot arm

Horizontal three-link open-chain manipulator (Figure 3.13) is the base of many industrial robots. With addition of a vertical moving end effector, the configuration is called SCARA (Selective Compliance Assembly Robot Arm) manipulator. It is developed in 1978 by Professor Hiroshi Makino at Yamanashi University. It has a high dexterity in the horizontal plane, which makes it suitable for manufacturing purposes.

Since this configuration has degree of freedom only in the horizontal plane, its dynamics does not involve destabilizing gravitational external moments. Developed fault tolerant control structure is applied to this configuration, in order to demonstrate the applications on systems with higher dimensions. Similar to the two-link case, free-swing type faults will be addressed.

#### 3.5.2.1 Problem formulation

Again the system dynamics is governed by the Lagrangian form for the robotic manipulators. It is repeated below for convenience.

$$H(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + G(q, \dot{q}) = \tau \tag{3.4}$$

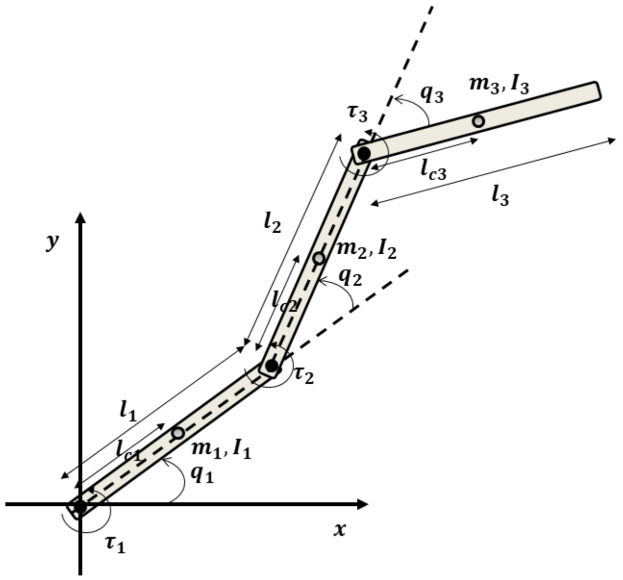


Figure 3.13: Vertical two-link robot manipulator system

Generalized position vector $q = (q_1, q_2, q_3)$ has three elements, corresponding to rotation angles as shown in 3.13. The system matrices $H$, $C$, $G$ and $\tau$ can be written as [93],

$$H(q) = \begin{bmatrix} \alpha + \beta + 2\gamma \cos q_2 & \beta + \gamma \cos q_2 & \delta \\ \beta + \gamma \cos q_2 & \beta & \delta \\ \delta & \delta & \delta \end{bmatrix}$$

$$C(q, \dot{q}) = \begin{bmatrix} -\gamma \sin q_2 \dot{q}_2 & -\gamma \sin q_2 \left(\dot{q}_1 + \dot{q}_2\right) & 0 \\ \gamma \sin q_2 \dot{q}_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C'(q, \dot{q}) = \begin{bmatrix} \gamma \sin q_2 \dot{q}_2 & -\gamma \sin q_2 \dot{q}_1 & 0 \\ \gamma \sin q_2 \left(\dot{q}_1 + \dot{q}_2\right) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 \end{bmatrix}^T$$

Since the base is assumed rigid, external gravitational moments are zero and therefore $G = 0$.

Parameters $\alpha, \beta, \gamma$ and $\delta$ are defined in terms of system parameters as,

$$\begin{aligned}
\alpha &= I_1 + m_1 l_{c1}^2 + m_2 l_1^2 + m_3 l_1^2 \\
\beta &= I_2 + I_3 + m_2 l_{c2}^2 + m_3 l_2^2 \\
\gamma &= m_2 l_1 l_{c2} + m_3 l_1 l_2 \\
\delta &= I_3
\end{aligned}$$

System parameters are shown in Figure 3.13.

#### 3.5.2.2 Design of the fault detection algorithm

State estimator can be formulated similar to the two-link case. Again $C'$ term is used in order to decouple the fault signal, with the following set of states: :

$$x = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & H_{11} & H_{12} & H_{13} \\ 0 & 0 & 0 & H_{21} & H_{22} & H_{23} \\ 0 & 0 & 0 & H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

Same general formulation (3.16) for state estimator of robotic manipulators is still valid:

$$\dot{\hat{x}} = \begin{bmatrix} \mathbf{0} & H^{-1} \\ \mathbf{0} & -C'H^{-1} \end{bmatrix} \cdot \hat{x} + \begin{bmatrix} \mathbf{0} \\ -G + \tau \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\hat{W}(t)^T \psi(x) \end{bmatrix} + A_s e_s$$

with the adaptation law,

$$\dot{\hat{W}} = \Gamma \cdot \text{Proj}\left(\hat{W}, -\psi(\hat{x}) \cdot e_s^T \cdot P_s\right)$$

where $P_s = P_s > 0$ solves the Lyapunov equation $A_s^T P_s + P_s A_s = -Q$ for $Q = \boldsymbol{I}$.

As in the two-link problem case (3.27), first four Chebyshev polynomials of first kind can be used as the basis set, but now for 3 different link positions:

$$\begin{aligned}
&\psi_0 = 1 \\
&\psi_1 = q_1 && \psi_7 = q_2 && \psi_{13} = q_3 \\
&\psi_2 = 2q_1^2 - 1 && \psi_8 = 2q_2^2 - 1 && \psi_{14} = 2q_3^2 - 1 \\
&\psi_3 = 4q_1^3 - 3q_1 && \psi_9 = 4q_2^3 - 3q_2 && \psi_{15} = 4q_3^3 - 3q_3 \\
&\psi_4 = \dot{q}_1 && \psi_{10} = \dot{q}_2 && \psi_{16} = \dot{q}_3 \\
&\psi_5 = 2\dot{q}_1^2 - 1 && \psi_{11} = 2\dot{q}_2^2 - 1 && \psi_{17} = 2\dot{q}_3^2 - 1 \\
&\psi_6 = 4\dot{q}_1^3 - 3\dot{q}_1 && \psi_{12} = 4\dot{q}_2^3 - 3\dot{q}_2 && \psi_{18} = 4\dot{q}_3^3 - 3\dot{q}_3
\end{aligned} \tag{3.28}$$

#### 3.5.2.3 Design of the fault mitigation algorithm

Gravitational effects are not present in horizontal manipulator cases. Therefore such systems are marginally stable and this eases the design process for $A_r$

Since $G = 0$, $\left.\frac{\partial f_1}{\partial x_2}\right|_{x_2=(r_2,0)}$ term given in (3.24) becomes $0$. For $\gamma_2$, $\|f_1(0, x_2)\|$ term can be evaluated as,

$$f_1(0, x_2) = \left.\begin{bmatrix} 0 \\ -H_{11}^{-1} C_{12} \dot{q}_2 \end{bmatrix}\right|_{(q_1,\dot{q}_1)=(0,0)}$$

#### 3.5.2.4 Simulation results

Free-swing fault on third joint (Loss of torque) case is studied as an example problem. In conjunction with Figure 3.13, numerical values used in the simulations are provided in Table 3.2,

At first sight, it might be wrongly assumed that since the system is marginally stable, the system might inherently tolerate an actuator fault. However, due to highly coupled nature of the system, fault in a single actuator affects all of the states. The situation is shown in Figure 3.14, where a free-swing fault on third link is introduced at $t = 5\,\mathrm{s}$, while the joints were following sinusoidal trajectories. Without any fault mitigation act, the system behaves erratically, as can be seen from the response of the system.

Fault mitigation strategy is chosen as introduction of perturbations on the second

Table 3.2: Simulated three-link robot arm's parameters

| $\boldsymbol{l_1}$ | $\boldsymbol{l_{c1}}$ | $\boldsymbol{m_1}$ | $\boldsymbol{I_1}$ |
|---|---|---|---|
| $1\,\mathrm{m}$ | $0.5\,\mathrm{m}$ | $1\,\mathrm{kg}$ | $0.0833\,\mathrm{kg\,m^2}$ |
| $\boldsymbol{l_2}$ | $\boldsymbol{l_{c2}}$ | $\boldsymbol{m_2}$ | $\boldsymbol{I_2}$ |
| $1\,\mathrm{m}$ | $0.5\,\mathrm{m}$ | $1\,\mathrm{kg}$ | $0.0833\,\mathrm{kg\,m^2}$ |
| $\boldsymbol{l_3}$ | $\boldsymbol{l_{c3}}$ | $\boldsymbol{m_3}$ | $\boldsymbol{I_3}$ |
| $1\,\mathrm{m}$ | $0.5\,\mathrm{m}$ | $1\,\mathrm{kg}$ | $0.0833\,\mathrm{kg\,m^2}$ |

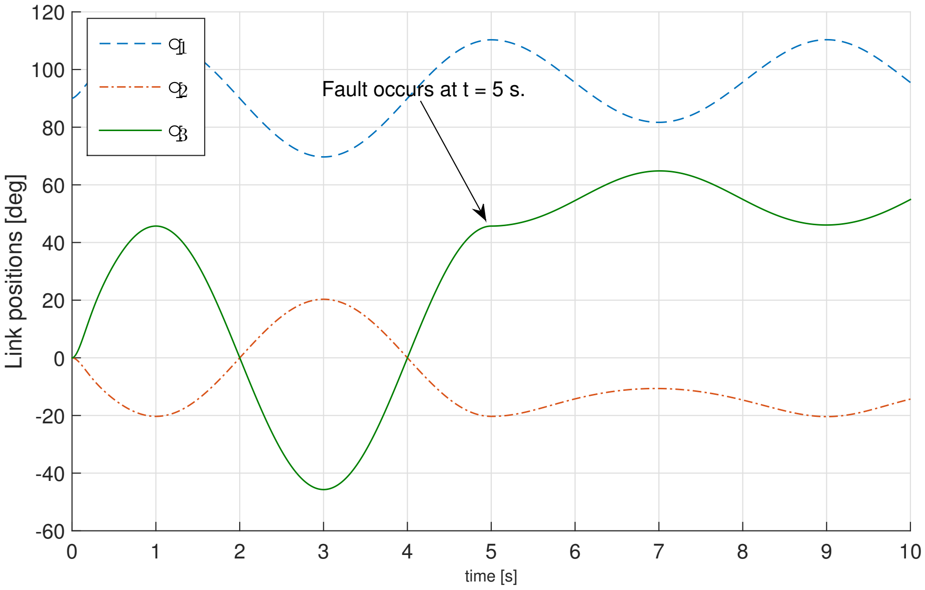


Figure 3.14: Link positions after the occurrence of a fault for the three-link robot arm problem simulation without any fault mitigation act

link, for the compensation of the fault on the third link. Therefore, the state matrices are adjusted so that the unactuated state is $x_1 = [q_1]$ and the actuated states are $x_2 = [q_1, q_2]$. The system matrices can be restructured as follows:

$$\begin{bmatrix} H_{33} & H_{31} & H_{32} \\ H_{13} & H_{11} & H_{12} \\ H_{23} & H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} C_{33} & C_{31} & C_{32} \\ C_{13} & C_{11} & C_{12} \\ C_{23} & C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ \tau_1 \\ \tau_2 \end{bmatrix} \qquad (3.29)$$

The structure of the nominal controller and the state estimator are the same as the ones used in two-link problem. Feedback gain matrices $K_v$ and $K_p$ are chosen such that the second order error dynamics has $\omega_n = 2\,\text{Hz}$ and $\xi = 0.7$.

$$K_p = \begin{bmatrix} 157.9 & 0 & 0 \\ 0 & 157.9 & 0 \\ 0 & 0 & 157.9 \end{bmatrix} \qquad K_v = \begin{bmatrix} 17.59 & 0 & 0 & & \\ 0 & 17.59 & 00 & 0 & 17.59 \end{bmatrix}$$

State estimator is constructed with,

$$A_s = \begin{bmatrix} -30 & 0 & 0 & 0 & 0 & 0 \\ 0 & -30 & 0 & 0 & 0 & 0 \\ 0 & 0 & -30 & 0 & 0 & 0 \\ 0 & 0 & 0 & -50 & 0 & 0 \\ 0 & 0 & 0 & 0 & -50 & 0 \\ 0 & 0 & 0 & 0 & 0 & -50 \end{bmatrix} \quad P_s = \begin{bmatrix} 0.010 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$$

First four Chebyshev polynomials of first kind are used as the basis set, as given in (3.28). Adaptive gain is $\Gamma_W = 1000$, $\epsilon_\theta$ is chosen as $0.01$.

For the fault mitigation algorithm, fault mitigation position is chosen as $(q_1, q_3) = (0°, -45°)$ and the following system is used as the reference model, which corresponds to a second order system with $\xi = 0.7$ and $\omega_n = 2\,\text{Hz}$.

$$A_{rm} = \begin{bmatrix} 0 & 1 \\ -157.9 & -17.59 \end{bmatrix} \qquad B_{rm} = \begin{bmatrix} 0 \\ 157.1 \end{bmatrix}$$

$A_r$ should be chosen in compliance with Conditions *C1* and *C2* of Theorem 2.4.1. As shown in previous section, $\left\| \left. \frac{\partial f_1}{\partial x_2} \right|_{x_2=(r_2,0)} \right\| = 0$. For $\|f_1(0, x_2)\|$, state definitions $x_1 = [q_1]$ and $x_2 = [q_1, q_2]$ with the structure in (3.29) can be used, which results in,

$$\|f_1(0, x_2)\| = \left| H_{33}^{-1} \left[C_{31} C_{32}\right] \dot{q}_2 \big|_{(q_3, \dot{q}_3)=(0,0)} \right|$$

Since $C_{31}$ and $C_{32}$ are both zero, for the horizontal three-link problem where the third actuator is faulty, $\|f_1(0, x_2)\|$ is zero. Therefore there is no restriction on $A_r$ and any Hurwitz matrix is sufficient for guaranteeing the stability of the system

$A_r$ is chosen in compliance with the specified form. Following matrix with the eigenvalues of $[-1.3820, -3.618]$ are chosen:

$$A_r = \begin{bmatrix} 0 & 1 \\ -5 & -5 \end{bmatrix} \qquad \alpha = 1 \quad \varepsilon = 0.5$$

Partial feedback linearized control law is used for application of the control signals. Following control gains are used, which corresponds to an error dynamics with $\xi = 0.7$ and $\omega_n = 1\,\mathrm{Hz}$.

$$K_1 = 8.7965 \qquad K_2 = 39.4784$$

Simulation is conducted for a hypothetical case where three-link arm is started at the position $(q_1, q_2, q_3) = (90°, 0°, 0°)$ and sinusoidal reference trajectories with frequency of $0.25\,\mathrm{Hz}$ are commanded to the system:

$$\begin{aligned} r_1 &= 90° + 20° \sin 2\pi t/4 \\ r_2 &= -20° \sin 2\pi t/4 \\ r_3 &= 45° \sin 2\pi t/4 \end{aligned}$$

Fault occurs at $t = 5sec.$. Simulation results are shown in Figure 3.15, Figure 3.16 and Figure 3.17 for the link positions, perturbation signal and the torque output of the controller respectively.

As required by the fault mitigation policy, an offset occurs on the position of the second link ($q_2$), in order to drive the system to the planned fault mitigation position of $(q_1, q_3) = (0°, -45°)$. A slight steady-state error of 3° occurs for the third link position.

### 3.6 Summary

In this chapter, application of the proposed algorithmic fault tolerant control structure on robotic manipulators is explained. Theoretical analyses of the Lagrangian form of the governing equations are developed and design steps are explained. Appropriate formulations for the signal and matrix bounds and design criteria are derived.

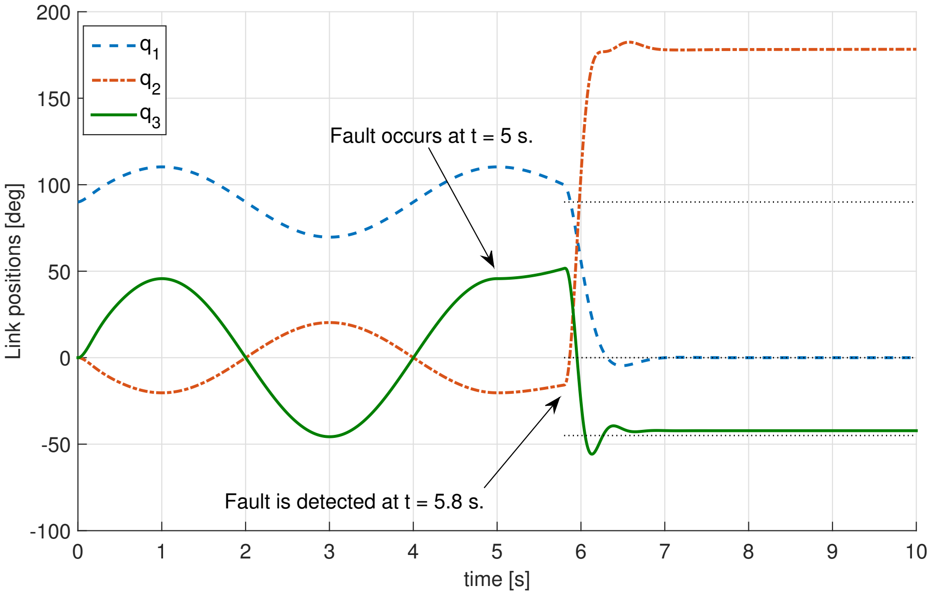


Figure 3.15: Link position for the three-link robot arm problem simulation

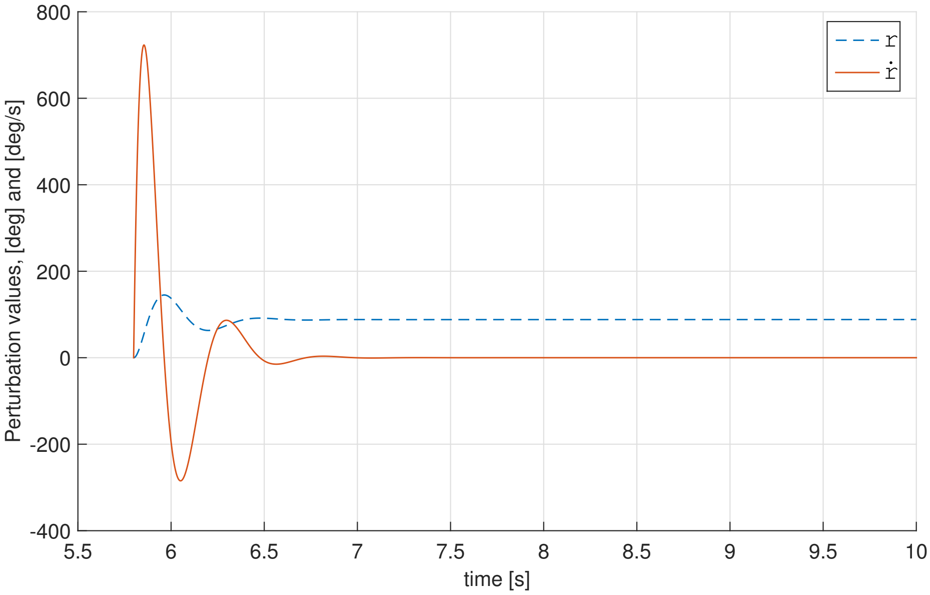


Figure 3.16: Perturbation on the second links position ($r$), for the compensation of fault on the third joint

Developed design methodology is applied on two benchmark problems; vertical oriented two-link robotic manipulator and horizontally located three-link robotic manipulator. As usual in nonlinear control systems, some problems are more suitable than the others. Bu the simulation results indicate that proposed strategy can be applied

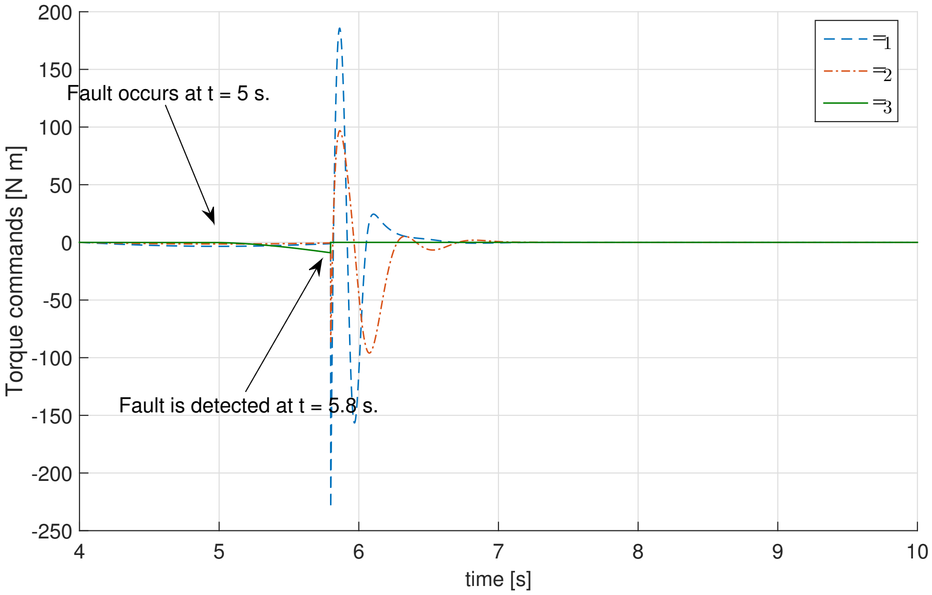


Figure 3.17: Applied torques for the three-link robot arm problem simulation

to robotic manipulator control systems and it has a potential as an alternative to existing control systems especially for underactuated systems.

# CHAPTER 4

# FAULT TOLERANT CONTROL OF QUADROTORS

Proliferation of low cost unmanned multirotor platforms lead developers to find increasingly new applications for such systems. This puts these platforms into roles that involve flight in challenging environments with complex and expensive payloads.

In order to cope with this usage, multirotor control research also diversified. One typical research area is in development of nonlinear controllers and trajectory generation algorithms for rapid maneuvering, typically for juggling and indoor flight applications, while another is development of control algorithms for systems with additional degrees of freedom such as vehicles with manipulator arms or other actuator mechanisms. Contrary to that, another interesting topic is maintaining stable flight with reduced number of rotors, for fault tolerance purposes or for novel vehicle design applications [106].

Such kind of a configuration is studied and impressively demonstrated in a TED talk by Mueller and D'Andrea [107]. They had shown that it is possible to align a typical quadrotor configuration around a desired vector in inertial coordinate frame using two propellers only. They found analytical periodic solutions of the dynamic system with reduced number of propellers, if the system is represented with so called the "reduced attitude", where the unit vector along the vertical axis of the quadrotor is used to represent the attitude. They have developed a control system based on linearization of this dynamics and applied it to single, two and tri-propeller cases [106]. Similar results are also achieved by Freddi, Lanzon and Longhi using two-stage feedback linearized controller for position and attitude [108, 109]. Other approaches for flight control with faulty propellers is finding suitable trajectories for following using three propellers [110] or using bounding functions on position commands to guarantee trajectory tracking [111].

Apart from above mentioned work, fault tolerant control research in multirotor platforms is mostly confined to compensation for partial thrust loss in propellers or rearrangement of the propeller thrusts in order to recover from loss of a propeller within overactuated systems such as hexacopters or octocopters [112, 113].

Attitude control is an essential part of any aerospace vehicle and therefore quadrotor attitude control problem is selected as an example application for the developed fault control strategy but the presented approach can be applied to other systems as well.

The proposed FTC structure against actuator faults for the quadrotor problem is based on maintaining the stability of the system through introduction of perturbations superposed on the main rotation rates of the actuated rotors. In this chapter, such an implementation will be explained. As explained in Section 1.3, complete implementation of a fault tolerant control system requires thorough analysis of various probable failure modes and testing of the control system for each failure type. With considerations on the length of the manuscript and in order to convey the essentials of the proposed methodology, only complete loss of single and two propellers will be discussed. But of analysis may be extended to include other types of failure.

This chapter is organized as follows. After introduction of the quadrotor control problem in Section 4.1, the fault diagnosis algorithm is derived in Section 4.3. Dynamics of the attitude control with reduced number of propellers are discussed in Section 4.4. After that, fault mitigation methodology is explained in Section 4.5. Performance of the control system is demonstrated using simulation results in Section 4.6, through simulation results on the selected cases. Final comments are given in Section 4.7.

## 4.1 Quadrotor control problem

Like many moving vehicles, quadrotor is an underactuated system [95]. Control in 6 dimensions with 12 states (position, velocity, attitude angles and angular rates) is achieved with manipulation of 4 propeller rotation rates. This is not as restrictive as it sounds, since unactuated internal states are stable, with obvious exception of altitude.

Nonlinear controllers with trajectory generation algorithms are required for rapid maneuvering but for near hover flight conditions, which is enough for many applications, position tracking is possible even with simple PD type controllers. This is usually achieved through employing a slow outer control loop for position tracking and an inner stabilization loop for control of the 3 angular rates $(p, q, r)$ and the total thrust of the propellers [114]. This structure is shown in Figure 4.1.

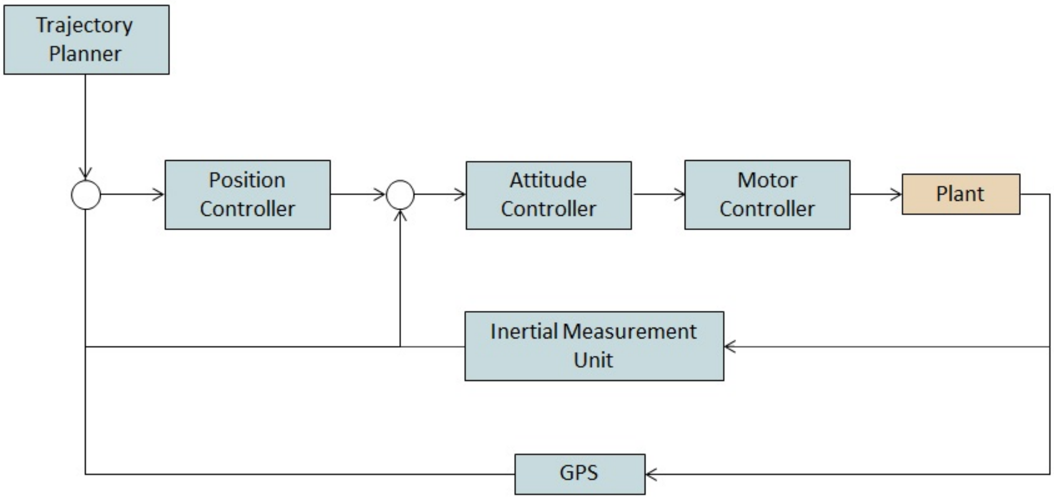


Figure 4.1: Elements of a typical quadrotor control system

Three control signals for stabilization of the three angular directions, together with the condition that their summation is equal to the specified thrust, four propellers can be fully mapped. However, once a propeller is lost, it is not possible to satisfy all four conditions. With the total thrust constraint, two propeller are remained for attitude control. However, Byrnes and Isidori have shonw that it is not possible to stabilize the full state of a spacecraft with two actuators [115].

In order to find a suitable sub-space of the attitude dynamics, a common approach in the literature is giving up on control of the yaw channel, so that the vehicle can freely rotate around its vertical axis [116]. With the control of pitch and roll, it is possible to direct the vertical axis of the vehicle at a certain orientation. This pointing direction is called the “Reduced attitue” and this vector can be controlled using three or less propellers. Therefore control with less than four propellers becomes a problem of finding stable sub-manifolds of the reduced attitude dynamics and existing work on the subject mainly focuses on this aspect of the problem.

Reduced attitude dynamics is also used in the developed control structure. But before the explanation of the proposed methodology, first the equations of motion and general attitude stabilization problem are discussed in the following sections.

### 4.1.1 Equations of motion

Equation of motion of a rigid body in 6 dimensions can be written using Newton's 2nd law of motion for the translational movement and Euler's equation for the rigid body rotations as:

$$\begin{aligned} m \cdot \ddot{x} =& F \\ I \cdot \dot{\omega}^B =& -\omega^B \times I \cdot \omega^B + M \end{aligned}$$

where $F$ is the vector of external forces and $M$ is the vector of external moments on the system, with respect to an inertial reference frame. $\omega^B$ is the angular velocity of the body, in body coordinate frame and $\ddot{x}$ is the acceleration in the inertial reference frame. This set of equations is universal and valid for any rigid body (With obvious limitations related to classical mechanics). Mathematical modeling comes into play for representation of external forces and moments specific to the system under study. For multirotor systems, it is common to assume that generated rotor force and torque are proportional to the square of the propeller speed as [114],

$$f_i = \kappa_f \cdot \omega_i^2 \qquad \tau_i = \kappa_\tau f_i = \kappa_\tau \cdot \kappa_f \cdot \omega_i^2$$

where $\kappa_f$ is the force constant and $\kappa_\tau$ is the torque constant of the propeller.

Modelling of other secondary affects such as blade flapping or rotational drag of propellers varies slightly in the literature. Modeling approach of Mueller and D'Andrea [106] are followed in this work and equations of motion for quadrotors are presented in (4.1), with axis definitions shown in Figure 4.2.

$$
\begin{aligned}
I_{xx}^{T}\cdot\dot{p} =& -\left(I_{zz}^{T}-I_{xx}^{T}\right)\cdot q\cdot r+\kappa_f\cdot l\cdot\left(\omega_2^2-\omega_4^2\right)\\
&-I_{zz}^{P}\cdot q\cdot\sum_i\omega_i-\kappa_{dxx}\cdot p\cdot\left\|\omega^B\right\|\\
I_{xx}^{T}\cdot\dot{q} =& \left(I_{zz}^{T}-I_{xx}^{T}\right)\cdot p\cdot r+\kappa_f\cdot l\cdot\left(\omega_3^2-\omega_1^2\right)\\
&+I_{zz}^{P}\cdot p\cdot\sum_i\omega_i-\kappa_{dxx}\cdot q\cdot\left\|\omega^B\right\|\\
I_{zz}^{T}\cdot\dot{r} =& \kappa_\tau\cdot\kappa_f\cdot\left(\omega_1^2-\omega_2^2+\omega_3^2-\omega_4^2\right)\\
&-\kappa_{dzz}\cdot r\cdot\left\|\omega^B\right\|\\
m\cdot a^B =& \kappa_f\cdot\left(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2\right)-\boldsymbol{R}_B^{E^{-1}}\cdot m\cdot g
\end{aligned}
\tag{4.1}
$$

where superscript $B$ indicates body coordinate frames with $\omega^B = (p, q, r)$, while $\omega_i$ represent the rotation rate of the propeller $i$. $\boldsymbol{R}_B^E$ is the directional cosine matrix from body frame to inertial frame. $l$ is the distance between the rotor axis and the center of the quadrotor. Inertial moment tensor $I^T$ is the total moment of inertia of the quadrotor, while $I^P$ is the moment of inertia of a single propeller. Quadrotor is assumed symmetric so that $I_{yy} = I_{xx}$. $\kappa_d$ terms represent the rotational drag of the propellers and again due to symmetry, $\kappa_{dxx} = \kappa_{dzz}$.

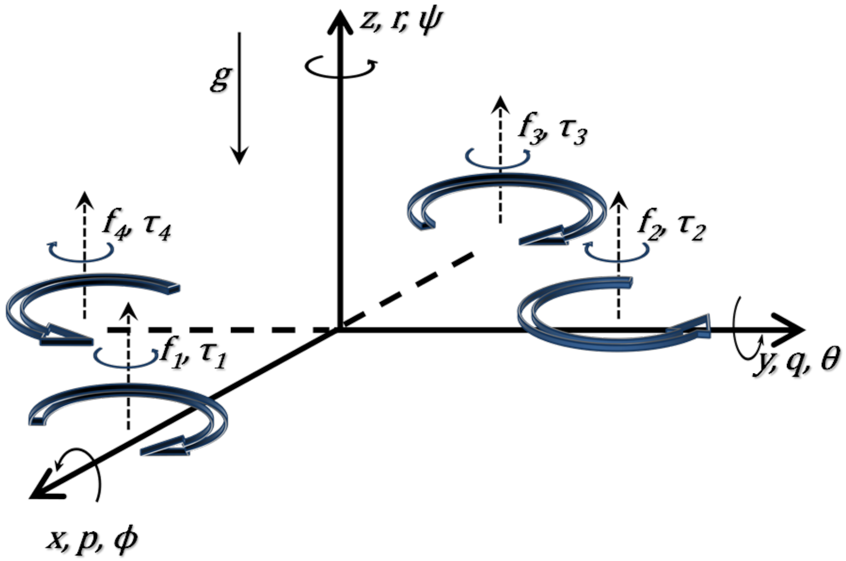


Figure 4.2: Axis and direction definitions for the quadrotor problem.

### 4.1.2 Attitude stabilization

System of equations (4.1) is a high fidelity model of a quadrotor system. It is suitable for nonlinear simulations but a simpler model is more appropriate for development of control algorithms.

Secondary effects in (4.1) can be neglected for $I^P \ll I^T$ and $\left\| \omega^B \right\| \ll \omega^2$. Also, yaw damping moment can be assumed as linearly dependent with the yaw rate $r$ so that $\kappa_{dzz} \cdot r \cdot \left\| \omega^B \right\| \approx \gamma r$. With these simplifications, system dynamics given in (4.2) can be used for control system design purposes.

$$\begin{aligned}
I_{xx}^B \cdot \dot{p} =& - \left(I_{zz}^T - I_{xx}^T\right) \cdot q \cdot r + l \cdot (f_2 - f_4) \\
I_{xx}^B \cdot \dot{q} =& \left(I_{zz}^T - I_{xx}^T\right) \cdot p \cdot r + l \cdot (f_3 - f_1) \\
I_{zz}^B \cdot \dot{r} =& \gamma r + \kappa_\tau \cdot (f_1 - f_2 + f_3 - f_4) \\
m \cdot a^B =& (f_1 + f_2 + f_3 + f_4) - \boldsymbol{R}_B^{E^{-1}} \cdot m \cdot g
\end{aligned}$$

or

$$\begin{aligned}
\dot{p} =& \frac{-\left(I_{zz}^T - I_{xx}^T\right)}{I_{xx}^B} \cdot q \cdot r + \frac{l}{I_{xx}^B} \cdot (f_2 - f_4) \\
\dot{q} =& \frac{\left(I_{zz}^T - I_{xx}^T\right)}{I_{xx}^B} \cdot p \cdot r + \frac{l}{I_{xx}^B} \cdot (f_3 - f_1) \\
\dot{r} =& -\gamma r + \frac{\kappa_\tau}{I_{zz}^B} \cdot (f_1 - f_2 + f_3 - f_4) \\
a^B =& \frac{1}{m} (f_1 + f_2 + f_3 + f_4) - \boldsymbol{R}_B^{E^{-1}} \cdot g
\end{aligned} \tag{4.2}$$

These set of equations are still highly nonlinear, but they are tractable and different control systems can be designed with. But another difficulty in attitude stabilization is the definition of angular errors in three dimensions [117] that can be used in the formulation of the control law.

For small angular errors, error vector defined from Euler angles can be used such as $e_R = \begin{bmatrix} \Delta\phi & \Delta\theta & \Delta\psi \end{bmatrix}^T$. However for large angular errors, a more complex definition is required. This can be achieved with the following error vector definition [114].

$$e_R \times = 1/2 \cdot \left( \boldsymbol{R}_{\boldsymbol{d}}^T \cdot \boldsymbol{R}_B^E - \boldsymbol{R}_B^{E^T} \cdot \boldsymbol{R}_{\boldsymbol{d}} \right)$$

where $R$ is the instantaneous directional cosine matrix from body frame to inertial frame and $\boldsymbol{R_d}$ is the desired attitudes directional cosine matrix. "$\times$" operator denotes the skew-symmetric form of a vector. It is defined for any vector $v \in \mathbb{R}^3$ as,

$$(v\times) = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

It should be noted that this expression is equivalent to $e_R = \begin{bmatrix} \Delta\phi & \Delta\theta & \Delta\psi \end{bmatrix}^T$ for small angles.

Once error vector $e_R$ is calculated, control signal can be generated using a PD controller,

$$u_R = -k_R \cdot e_R - k_\omega \cdot e_\omega \tag{4.3}$$

where derivative term is calculated $e_\omega = \omega_d - \omega^B$.

Once control signal $u_R \in \mathbb{R}^3$ is calculated, together with an acceleration constraint $\sum f_i = F_d$, four signals can be mapped to four propellers $f_i$. $F_d$ is the desired total force magnitude that can be calculated from the required acceleration.

This transformation can be found from (4.2) as a matrix equality:

$$\begin{bmatrix} F_d \\ u_{R1} \\ u_{R2} \\ u_{R3} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & l & 0 & -l \\ -l & 0 & l & 0 \\ \kappa_\tau & -\kappa_\tau & \kappa_\tau & -\kappa_\tau \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

or with taking the inverse of the transformation matrix,

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 0 & -\frac{2}{l} & \frac{1}{\kappa_\tau} \\ 1 & \frac{2}{l} & 0 & -\frac{1}{\kappa_\tau} \\ 1 & 0 & \frac{2}{l} & \frac{1}{\kappa_\tau} \\ 1 & -\frac{2}{l} & 0 & -\frac{1}{\kappa_\tau} \end{bmatrix} \begin{bmatrix} F_d \\ u_{R1} \\ u_{R2} \\ u_{R3} \end{bmatrix}$$

However, once a propeller is failed, this mapping can not be done. Reduced attitude is a useful representation of the attitude for such kind of cases [116, 117]. It is defined as the pointing direction of the attitude (positive $z$ axis in Figure 4.2) in inertial coordinate system. Therefore it can be formulated as,

$$\Gamma = \boldsymbol{R}_B^E \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Since the cross product defines a vector perpendicular to the input vectors with magnitude of sine of the angle between them, error vector for reduced attitude can be calculated as,

$$e_R = \Gamma \times \Gamma_d$$

where, $\Gamma_d \in \mathbb{R}^3$ is the desired reduced attitude.

Proportional-Derivative control structure given in (4.3) can still be used with reduced attitude [117]. Although the calculated control input is still $u_R \in \mathbb{R}^3$, yaw channel control input ($\Gamma_3$) becomes zero and calculated control signal is independent of the yaw angle.

This error formulation is mainly used in control system design methods available in the literature. But for the proposed methodology, a little bit more detail about the reduced attitude dynamics should be discussed.

## 4.2 Reduced attitude dynamics

Reduced attitude can be interpreted as the pointing direction of a system. For quadrotors, it is taken as the vertical direction ($z$ in Figure 4.2) that is parallel to the rotor axes but it can be taken as any vector. For example, pointing direction of an antenna on a satellite can also be manipulated using reduced attitude.

With this definition, reduced attitude can be represented as a unit vector on a sphere [116]. Consider two such vectors $(p, q)$, one representing the vehicles attitude and the other one is the desired reduced attitude. Both of these vectors can be represented as points on the same sphere (Same coordinate system) with appropriate metrics. The situation is shown in Figure 4.3.

Distance between two vectors can be taken as the angle between them on the plane

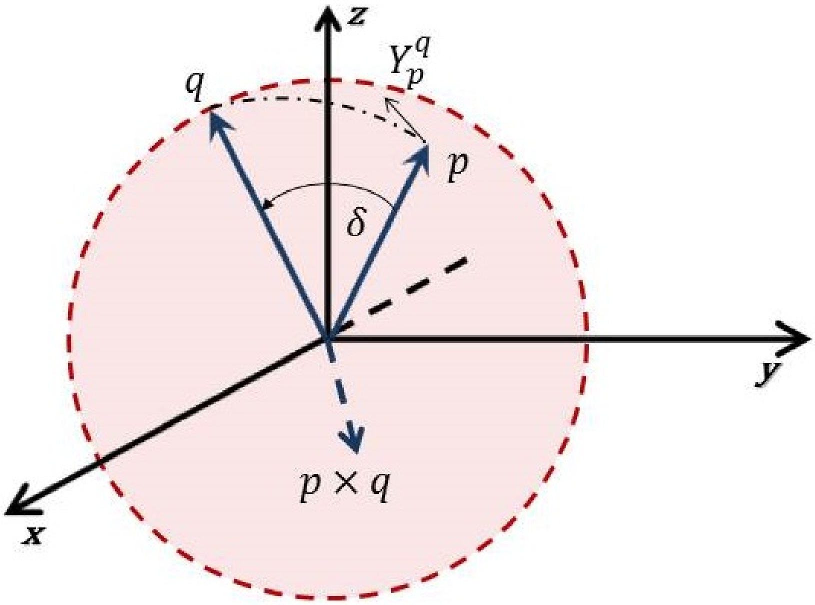


Figure 4.3: Reduced attitude and related parameters.

formed with these vectors. This can be formulated as,

$$dist(p,q) = \delta = \arccos\left(\langle p, q\rangle\right)$$

where $\langle \cdot\,, \cdot \rangle$ is the scalar or dot product and for vector in $\mathbb{R}^{3\times1}$, it can be calculated as $\langle u, v\rangle = u^T v$.

Geodesic direction in $p$ towards $q$ can be formulated as,

$$Y_p^q = vers\left(p \times q\right) \times p \tag{4.4}$$

where versor operator $vers$ basically normalizes the input vector as $vers\left(x\right) = x/\left\|x\right\|$.

With these definitions, time rate of change of distance can be formulated as [116]:

$$\frac{d\delta}{dt} = -\langle \dot{p}, Y_p^q \rangle \tag{4.5}$$

Now consider the reduced attitude of the system as $p$ and let the desired reduced attitude vector $q$ be the positive $z$ axis in inertial frame:

$$p = R_B^E \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad q = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Since unit vector in $z$direction is constant, time rate of change of $p$ can be derived as,

$$\dot{p} = \dot{R}_B^E \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = R_B^E \left( \omega^B \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right)$$

where $\dot{R}_B^E = R_B^E \left( \omega^B \times \right)$ relation from attitude kinematics is used. For $\omega^B = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T$, $\dot{p}$ can be formulated as,

$$\dot{p} = R_B^E \begin{bmatrix} \omega_2 \\ -\omega_1 \\ 0 \end{bmatrix} \tag{4.6}$$

Similarly, $Y_p^q$ can be calculated for $q = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ using 4.4 as,

$$Y_p^q = \frac{1}{\sqrt{p_1^2 + p_2^2}} \begin{bmatrix} p_2 \\ -p_1 \\ 0 \end{bmatrix} \times \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} \frac{-p_1 p_3}{\sqrt{p_1^2+p_2^2}} \\ \frac{-p_2 p_3}{\sqrt{p_1^2+p_2^2}} \\ \sqrt{p_1^2 + p_2^2} \end{bmatrix} \tag{4.7}$$

where $p = \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix}^T$

Time rate of change of distance between the reduced attitude of the system and the vertical axis can be formulated using (4.6) and (4.7) with (4.8):

$$\frac{d\delta}{dt} = -\langle \dot{p}, Y_p^q \rangle = -\left\langle R_B^E \begin{bmatrix} \omega_2 \\ -\omega_1 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{-p_1 p_3}{\sqrt{p_1^2+p_2^2}} \\ \frac{-p_2 p_3}{\sqrt{p_1^2+p_2^2}} \\ \sqrt{p_1^2 + p_2^2} \end{bmatrix} \right\rangle$$

Definition of inner product can be used for further simplification since:

$$\langle u, v \rangle = u^T v \rightarrow \langle Au, v \rangle = u^T A^T v$$

and

$$\left\langle R_B^E \begin{bmatrix} \omega_2 \\ -\omega_1 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{-p_1 p_3}{\sqrt{p_1^2+p_2^2}} \\ \frac{-p_2 p_3}{\sqrt{p_1^2+p_2^2}} \\ \sqrt{p_1^2 + p_2^2} \end{bmatrix} \right\rangle = \begin{bmatrix} \omega_2 \\ -\omega_1 \\ 0 \end{bmatrix}^T R_B^{E^T} \begin{bmatrix} \frac{-p_1 p_3}{\sqrt{p_1^2+p_2^2}} \\ \frac{-p_2 p_3}{\sqrt{p_1^2+p_2^2}} \\ \sqrt{p_1^2 + p_2^2} \end{bmatrix}$$

Let vector $p'$ be,

$$\begin{bmatrix} p'_1 \\ p'_2 \\ p'_3 \end{bmatrix} = R_B^{E^T} \begin{bmatrix} \frac{-p_1 p_3}{\sqrt{p_1^2+p_2^2}} \\ \frac{-p_2 p_3}{\sqrt{p_1^2+p_2^2}} \\ \sqrt{p_1^2+p_2^2} \end{bmatrix}$$

With this definition, time derivative of distance can be calculated

$$\frac{d\delta}{dt} = \omega_1 p'_2 - \omega_2 p'_1 \tag{4.8}$$

As expected, rate of change of distance is independent of the yaw rate ($\omega_3$). This relation will be used in construction of the control system in Section 4.5.

### Reduced attitude stabilization

A Proportional plus derivative control law for reduced attitude control is derived by Bullo, Murray and Sarti for a general attitude control problem [116], using the parameters defined in this section. It is more general than the control laws presented in previous section and therefore it is worth mentioning:

Consider the current reduced attitude vector $p_0$ and the desired reduced attitude vector $q$. Given positive control gain $k_p$ and positive definite matrix $K_d$, following control torque renders the equilibrium point $(p = q, w_1 = 0, w_2 = 0)$ stable from any initial condition, provided that $p_0$ is not the antipodal vector of $q$, *i.e.*,$p_0 \neq -q$.

$$\tau = k_p dist(p_0, q) \begin{bmatrix} -\langle Y_{p_0}^q, p_2 \rangle \\ \langle Y_{p_0}^q, p_1 \rangle \end{bmatrix} - K_d \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \tag{4.9}$$

where $p_1 = R_B^E \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$ and $p_2 = R_B^E \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$.

The stability is guaranteed for all $k_p$ and angular velocity vector $\omega$ such that,

$$k_p > \frac{\langle \omega, I\omega \rangle}{\pi^2 - dist\left(p_0, q\right)^2}$$

These relations are derived for spacecraft attitude control problem and any disturbance torque such as aerodynamics or rotors are disregarded. But in quadrotor problem, such affects are stabilizing and therefore their presence does not pose any problem.

## 4.3 Fault diagnosis algorithm for quadrotors

Formulation of the fault diagnosis algorithm, using state estimator structure is straightforward. Disturbances on each of the four propellers can be estimated in conjunction with (4.2), using the measurements of states $(p, q, r, a_z^B)$. However, since thrust of the propellers are always positive, loss of each propeller has a unique disturbance character. For example, for the configuration shown in Figure 4.2, loss of thrust in second propeller causes a negative net moment around $X$ axis. Or, if first and third propellers are lost simultaneously, negative net moment occurs around $Z$ axis. Using this relation, fault can be diagnosed using only angular rate measurements.

State estimator can be formulated as follows,

$$\begin{bmatrix} \dot{\hat{p}} \\ \dot{\hat{q}} \\ \dot{\hat{r}} \end{bmatrix} = \begin{bmatrix} 0 & -ar & 0 \\ ar & 0 & 0 \\ 0 & 0 & -\gamma \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} \frac{l}{I_{xx}^T} \cdot (f_2 - f_4) \\ \frac{l}{I_{xx}^T} \cdot (f_3 - f_1) \\ \frac{\kappa_\tau}{I_{zz}^T} \cdot (f_1 - f_2 + f_3 - f_4) \end{bmatrix} + \widehat{D} + A_s e_s \quad (4.10)$$

where $\widehat{D} \in \mathbb{R}^{3\times 1}$ is the disturbance vector and

$$a = \frac{\left(I_{zz}^T - I_{xx}^T\right)}{I_{xx}^T}$$

Error vector is $e_s = \begin{bmatrix} \hat{p} & \hat{q} & \hat{r} \end{bmatrix}^T - \begin{bmatrix} p & q & r \end{bmatrix}^T$ and $A_s$ is a Hurwitz gain matrix for the estimator.

The disturbance vector can be approximated with projection on a basis $\psi_{1..N}(\hat{x})$ as

$$\widehat{D}_i \approx \sum_{j=1}^{N} W_{ij} \cdot \psi_j(\hat{x}) \quad (4.11)$$

where $N$ is the number of basis functions and subscript $i$ indicate the element of the vector. Similar to robotic manipulator problem (Section 3.2), Chebyshev polynomials can be used as the basis set.

The weights of the basis functions can be updated with the error dynamics as,

$$\dot{\widehat{W}} = \Gamma \cdot \text{Proj}\left(\widehat{W}, -\psi(\hat{x}) \cdot e_s^T \cdot P_s\right) \quad (4.12)$$

with $P_s = P_s > 0$ solves the Lyapunov equation $A_s^T \cdot P_s + P_s \cdot A_s = -Q$ for arbitrary $Q > 0$. Proj is the projection operator, commonly used in adaptive controllers in order to bound the estimated gains. The formulation of the operator is given in Section 2.6 as (2.20).

With the estimated disturbance vector, faulty propeller can be identified as explained above. Determination of the faulty propeller derived from the sign of the elements of the disturbance vector is summarized in Table 4.1. With this parameter set, auxiliary parameter $a$ can be calculated as:

$$a = \frac{\left(I_{zz}^{T} - I_{xx}^{T}\right)}{I_{xx}^{T}} = 0.9259$$

Table 4.1: Fault diagnosis from the disturbance vector

| **Fault Case** | **Sign of the Elements** | | |
|---|---|---|---|
| | $D_1$ | $D_2$ | $D_3$ |
| No Fault | $0$ | $0$ | $0$ |
| Propeller 4 is failed | $+$ | $0$ | $0$ |
| Propeller 2 is failed | $-$ | $0$ | $0$ |
| Propeller 1 is failed | $0$ | $+$ | $0$ |
| Propeller 3 is failed | $0$ | $-$ | $0$ |
| Propeller 2 and 4 are simultaneously failed | $0$ | $0$ | $+$ |
| Propeller 1 and 3 are simultaneously failed | $0$ | $0$ | $-$ |
| Propeller 1 and 4 are simultaneously failed | $+$ | $+$ | $0$ |
| Propeller 2 and 3 are simultaneously failed | $-$ | $-$ | $0$ |

## 4.4 Attitude Control with reduced number of propellers

Before the derivation of the FTC algorithm, it is instructive to discuss the dynamics of the system with reduced number of propellers.

Since all rotors are located on the $x - y$ plane, numbering of the rotors can be changed with rotation of the axis. Therefore loss of the fourth propeller case in Figure 4.2 will be discussed without losing any generality, while other three propellers produce thrust.

Equation of motion of this system is repeated below for convenience.

$$\begin{aligned}\dot{p} &= -aqr + cf_2 \\ \dot{q} &= apr + c\left(f_3 - f_1\right) \\ \dot{r} &= -\gamma r + b\left(f_1 - f_2 + f_3\right)\end{aligned} \tag{4.13}$$

where $a, b, c, \gamma > 0$.

Thrust of the second propeller ($f_2$) is always positive and it creates a positive signed disturbance on the roll channel ($p$). If the pitch and yaw rates are zero, then it is impossible to stabilize the $p$ dynamics with a nonzero $f_3$. But for nonzero $q \cdot r$ term, stabilization of roll dynamics might be possible. For example it is possible to find an equilibrium solution for motion with constant $q$ where $f_3 = f_1$ and $f_2 = f_1 + f_3$ so that $r \approx constant$ (for $\gamma \approx 0$). Actually, this kind of solutions are proposed by Mueller and D'Andrea for the quadrotor motion with three propellers [107].

Now as an alternative, consider again the motion under constant yaw rate ($r = \bar{r} > 0$). If $q$ involves a perturbation, $p$ dynamics can be written as:

$$\begin{aligned}\dot{p} &= -a\bar{r}\left(q + q'\right) + cf_2 \\ \dot{q} &= a\bar{r} + c\left(f_3 - f_1\right)\end{aligned}$$

If the magnitude of the perturbation is adjusted so that $-rq' + cf_2 = 0$, $p$ dynamics would be stabilized. Proposed FTC algorithm can generate such kind of signals.

If another propeller is lost, there are two possibilities. If the second propeller is lost, it is not possible to achieve a zero yaw rate, sine $f_1 + f_3$ is always positive. Yaw rate would settle at a constant rate of $\bar{r} = \frac{b}{\gamma}\left(f_1 + f_3\right)$ and $p$ dynamics become unactuated. Mueller and D'Andrea had shown that it is possible to "point" the $z$ axis of the system under this condition, using the reduced attitude dynamics. Proposed FTC algorithm can generate a perturbation signal on $q$ to achieve the same goal, with the advantage of not requiring the reformulation of the original control system.

The other possibility is losing the third propeller. The total thrust of first and second propellers should counter the weight of the system. Suppose that same thrust is applied on both first and second propeller and control inputs are applied as deviations from the second propeller so that $f_1 = f_0$ and $f_2 = f_0 + u$ with $f_0 > 0$. Then the attitude dynamics can be written as,

$$\begin{aligned}\dot{p} &= -arq + d\\ \dot{q} &= arp - d + u\\ \dot{r} &= -\gamma r - \frac{b}{c}u\end{aligned}$$

with disturbance signal $d > 0$. In this situation, yaw rate is not automatically induced but control inputs $u(t)$ creates both pitch rate and yaw rate commands. Equilibrium is possible for only nonzero $(p, q, r)$ attitude rates.

From the system dynamics point of view, flight with a single propeller is not much a different condition than flight with two co-rotating propellers. Again a yaw rate is created and again a positive moment around $x$ or $y$ axis is created, depending on the remaining propeller. Therefore, the solution would be very similar to the two co-rotating propellers case.

Although it is possible to guess the existence of a solution, from the discussions related to physics of the problem, it is another thing to find a control system that implements such results [1]. In the next section, formulation of the controller, based on the proposed FTC algorithm, is presented. Simulation results for the above discussed cases are provided in the subsequent sections.

### 4.5 Fault tolerant controller for quadrotors

Loss of propeller causes the disappearance of the force and moments created by the faulty propeller. This kind of fault is in compliance with the developed fault mitigation strategy of zeroing the effect of the faulty actuator and introduction of perturbations on the healthy ones in order to stabilize the overall system. Once a propeller is lost, FTC should start injecting the state perturbations on the controller, without taking any other action.

As shown in Figure 4.1, a typical flight control system includes outer trajectory planing and position control loops. The output of these loops is basically three Euler angles for the desired angular position of the vehicle relative to earth and required acceleration.

---

[1] To be honest, I personally wouldn't deduct the existence of such solutions, if they had not been shown by Mueller and D'Andrea [106, 107]

These guidance commands are fed to the control system that generate thrust commands (in fact, rotor revolution rates) in compliance with attitude dynamics given in (4.2). Fault tolerant control algorithm is developed to introduce fault mitigation perturbations on the attitude rates. Therefore calculated perturbations are added to the attitude rate commands within the attitude controller part of the control system. The overall flight control system architecture with the proposed fault tolerant control elements is shown in Figure 4.4.

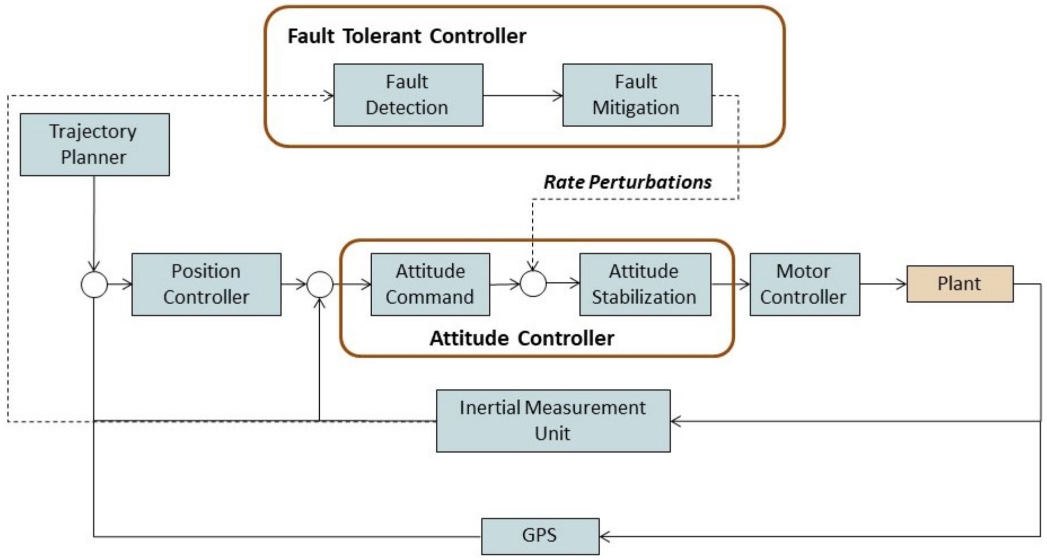


Figure 4.4: Quadrotor flight control system architecture, including the FTC algorithm

As explained in Section 4.4, loss of a single propeller results in an asymmetric moment around the axis of the healthy propellers and system inevitably gains a yaw rate. Actually, spinning systems gain gyroscopic stability and therefore admission of high yaw rates is not as bad it sounds. In fact, if the thrust of two healthy propellers are enough to maintain the desired motion, than it would be a good idea to stop the propeller that is opposite to the failed propeller so that the disturbance moment would disappear and the configuration falls back to the two propeller case. Therefore for the design consideration of the FTC algorithm, single and double propeller cases are the same problem with different third propeller thrust values.

Assume that $4^{\text{th}}$ propeller is lost and let $f_{2,ftc}$ be the constant thrust that will be applied from the second propeller. Disturbance moment can be written as $d = cf_{2,ftc} > 0$ and the resulting attitude dynamics is,

$$\begin{aligned} \dot{p} &= -aqr + d \\ \dot{q} &= apr + c\left(f_3 - f_1\right) \\ \dot{r} &= -\gamma r + b\left(f_1 + f_3 - f_{2,ftc}\right) \end{aligned} \tag{4.14}$$

where $a, b, c, \gamma$ are positive constants that depends on system parameters as,

$$b = \frac{\kappa_\tau}{I_{zz}^T} \qquad c = \frac{l}{I_{xx}^T}$$

For $f_1, f_3 > 0$, it is not possible to achieve zero yaw rate. However, since yaw dynamics is first order, with $u_r = (f_1 + f_3 - f_{2,ftc})$ as input, its time dependent response for constant $u_r$ and with $r(t_0) = 0$ can be written as,

$$r(t) = \frac{b}{\gamma} u_r \left(1 - e^{-t\frac{\gamma}{b}}\right)$$

Therefore, for constant $u_r$, maximum yaw rate that the yaw dynamics can attain is,

$$r_{max} = \frac{b}{\gamma}\left(f_1 + f_3 - f_{2,ftc}\right) \tag{4.15}$$

$f_{2,ftc}$ is a constant thrust that will be applied after the detection of the fault. If the motor control inputs are calculated such that $f_1 + f_3 = constant$, than $u_r = constant$ and it would be guaranteed that the yaw rate is always bounded with this predetermined value. In practice, maximum motor rates are limited and therefore it is possible to calculate a maximum attainable value for yaw rate. However, there is an additional consideration that should be given to the calculation of motor force commands.

The control input on the pitch channel ($q$) is $u = c\left(f_3 - f_1\right)$ and this signal is calculated by the proposed fault tolerant control system. For each calculated value of $u$ and for constant predetermined value of $u_r$, force commands on the propellers $f_1$ and $f_3$ can be calculated with following constraints:

$$\begin{aligned} f_1 - f_3 &= -\frac{u}{c} \\ f_1 + f_3 &= f_{2,ftc} + u_r \end{aligned}$$

It is possible to find $f_1$ and $f_2$ that satisfies the above relations, but unfortunately for quadrotors, there is an additional constraint that $f_1, f_3 > 0$. Since the lowest value of

$f_3$ is zero, the first equation provides a limit on $u$ such that,

$$\max|u| \leq cf_{max} \tag{4.16}$$

where the propeller force limit be denoted as $f_{max}$.

On the other hand, choosing $u_r$ such that,

$$u_r \leq f_{max} - f_{2,ftc}$$

guarantees that second equation is satisfied even for $f_1$ or $f_3 = 0$. Using this relation in (4.15) results in a limit for maximum yaw rate as,

$$|r(t)| \leq \frac{b}{\gamma}\left(f_{max} - f_{2,ftc}\right) \tag{4.17}$$

In application, this limits should be calculated and the overall design of the control system should be consistent with respect to the actuator limit (4.16) and the corresponding bound on $r$, given in (4.17), which will be used for determination of $A_r$ in the proposed control system.

With this considerations, pitch and roll channels $(p, q)$ can be treated separately using reduced attitude representation and yaw rate $(r)$ can be considered as an external bounded signal. The main fault mitigation act is pointing the thrust axis against the gravity, so that the system maintains a stable reduced altitude. This can be formulated as the desired reduced attitude vector of $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. The dynamics of the distance between the current reduced attitude and the desired reduced attitude is given in Section 4.2.

With this respect, considering the controller design strategy that had been explained in Chapter 2, roll rate $(p)$ and the reduced attitude distance from the $z$ axis can be treated as the uncontrolled states $x_1$ and pitch rate $(q)$ becomes the controlled state $x_2$. With this definitions, interconnected system dynamics can be written as,

$$\begin{aligned}
\dot{x}_1 &= \begin{bmatrix} \dot{\delta} \\ \dot{\omega}_1 \end{bmatrix} = \begin{bmatrix} 0 & p_2' \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \\ \omega_1 \end{bmatrix} + \begin{bmatrix} -p_1' \\ -a\omega_3 \end{bmatrix} x_2 + \begin{bmatrix} 0 \\ d \end{bmatrix} \\
\dot{x}_2 &= \dot{\omega}_2 = a\omega_3 \cdot \omega_1 + u
\end{aligned} \tag{4.18}$$

In order to avoid confusion, $\omega^B = \begin{bmatrix}\omega_1 & \omega_2 & \omega_3\end{bmatrix}^T$ naming is preferred for the angular rates, instead of the $(p, q, r)$. Definitions of $\delta$, $p_1'$ and $p_2'$ are provided in Section 4.2.

Control inputs $u(t)$ are generated with creating an offset on the nominal propeller forces as,

$$u = c\,(f_3 - f_1)$$

### 4.5.1 Perturbation relation

Perturbations on the controlled pitch channel ($w_2$) can be calculated from the main FTC relation:

$$\epsilon\dot{r} = \alpha \cdot [f_1(x_1, r_2 + r) - f_{r1}(x_1, B_r \cdot r_2)] + A_r r$$

However, uncontrolled dynamics are in $x_1 \in \mathbb{R}^2$, while the controlled dynamics is in $x_2 \in \mathbb{R}^1$. Therefore it is not possible to directly map the calculated perturbations to the pitch rate perturbations. This can be achieved using a control matrix $B_r$ for application of the perturbations.

Let $q'$ be the applied perturbation on $x_2$. Then,

$$q' = B_r r$$

with $B_r \in \mathbb{R}^{2\times1}$ results in the desired formulation.

Hence the perturbation equation for the attitude control problem can be formulated as,

$$\begin{aligned}\epsilon\dot{r} &= \alpha \cdot [f_1(x_1, B_r\,(r_2 + r)) - f_{r1}(x_1, B_r r_2)] + A_r r\\ q' &= B_r r\end{aligned}$$

With the system model given in (4.18), it can be written that,

$$\begin{aligned}\epsilon\dot{r} &= \alpha \cdot \left\{ \begin{bmatrix}0 & p_2'\\ 0 & 0\end{bmatrix} \begin{bmatrix}\delta\\ \omega_1\end{bmatrix} + \begin{bmatrix}-p_1'\\ -a\omega_3\end{bmatrix} B_r r + \begin{bmatrix}0\\ d\end{bmatrix} - A_{rm} \begin{bmatrix}\delta\\ \omega_1\end{bmatrix} \right\} + A_r r\\ q' &= B_r r\end{aligned} \tag{4.19}$$

$B_r$ determines how the perturbations would affect the system. Main aim of the FTC controller is to make $\delta \to 0$. Therefore,

$$B_r = \begin{bmatrix}1 & 0\end{bmatrix}$$

would give priority to this channel.

As usual, $A_{rm}$ is the desired behaviour of the uncontrolled system

### 4.5.2 Stability of the boundary layer equation

The conditions for the stability of the boundary layer equation is analyzed through function $f_1(x_1, x_2)$ of the cascade dynamic system structure (2.3) given in Section 2.2. This function can be identified from (4.18) as,

$$f_1(x_1, x_2) = \begin{bmatrix} 0 & p_2' \\ 0 & 0 \end{bmatrix} x_1 + \begin{bmatrix} -p_1' \\ -a\omega_3 \end{bmatrix} x_2 + \begin{bmatrix} 0 \\ d \end{bmatrix}$$

The conditions *C1* and *C2* of Theorem 2.4.1, which is given in Section 2.4 can be evaluated as,

$$\begin{aligned} f_1(0, x_2) &= \begin{bmatrix} -p_1' \\ -a\omega_3 \end{bmatrix} x_2 + \begin{bmatrix} 0 \\ d \end{bmatrix} \\ \left.\frac{\partial f_1}{\partial x_2}\right|_{x_2=(r_2,0)} &= \begin{bmatrix} -p_1' \\ -a\omega_3 \end{bmatrix} \end{aligned}$$

The presence of a nonzero $d$ term violates the $f_1(0,0) = 0$ assumption, given as Assumption *A1* of Theorem 2.4.1. But as stated in Remark 2.4.6, this results in a shift in equilibrium point of $x_1$. As will be demonstrated in numerical simulations (Section 4.6), a nonzero $d$ term causes a constant roll angular rate $p$ and tilts the pointing angle of the quadrotor (Body $z$ axis).

The presence of $d$ does not change the stability properties of the system $\dot{y} = f_1(0, y)$ and since it is in the form of,

$$\dot{y} = f_1(0, y) = Ay$$

the bound on $\|f_1(0, x_2)\|$ can be calculated as,

$$\|f_1(0, x_2)\| = \left\| \begin{bmatrix} -p_1' \\ -a\omega_3 \end{bmatrix} x_2 \right\| = \left\| \begin{bmatrix} -p_1' \\ -a\omega_3 \end{bmatrix} \right\| \|x_2\| \tag{4.20}$$

Therefore $\gamma_2$ in Condition *C2* of Theorem 2.4.1 can be identified as $\gamma_2$. Actually, this is also equal to the $\left\| \frac{\partial f_1}{\partial x_2}\Big|_{x_2=(r_2,0)} \right\|$ and therefore the Conditions *C1* and *C2* of Theorem 2.4.1 can be simultaneously satisfied with,

$$\gamma_1 = \gamma_2 = \left\| \begin{bmatrix} -p_1' \\ -a\omega_3 \end{bmatrix} \right\| \tag{4.21}$$

The Frobenius norm is more appropriate for calculation of this norm:

$$\left\| \begin{bmatrix} -p_1' \\ -a\omega_3 \end{bmatrix} \right\|_P = \sqrt{tr\left( \frac{\partial f_1}{\partial x_2}\Big|_{x_2=(r_2,0)} \frac{\partial f_1}{\partial x_2}\Big|_{x_2=(r_2,0)}^T \right)} = \sqrt{p_1'^2 + (a\omega_3)^2}$$

Since $p_1'$ is an element of a unit vector, it is bounded by $1$. Also $a < 1$ for a general quadrotor configuration. Therefore following inequality holds:

$$\left\| \begin{bmatrix} -p_1' \\ -a\omega_3 \end{bmatrix} \right\|_P \leq \sqrt{1 + \omega_3^2} \leq \sqrt{1 + \omega_{3,max}^2}$$

$A_r$ can be chosen for the maximum attainable yaw rate $\omega_{3,max}$ using (4.17), such that $A_r$ that solves the Lyapunov equation $PA_r + A_r^T P + Q = 0$ with $Q$ as the identity matrix with appropriate dimensions and $\frac{1}{2\lambda_{max}(P)} > \gamma_1 = \gamma_2$.

Maximum eigenvalue of $A_r$ for different maximum yaw rates are calculated and the results are shown in Figure 4.5.

With this result, following comments can be made regarding the assumptions and conditions of the main theorem on the fault mitigation control structure Theorem 2.4.1.

- The quadrotor dynamics have a continuous parameter sets. Therefore assumption on Lipschitz continuity of the functions, Assumption *A1* of Theorem 2.4.1 are valid.
- Assumption *A2* of Theorem 2.4.1 is related to the controllability of $x_1$ dynamics with $x_2$ regarded as control input. This dynamics, $\dot{x}_1 = f_1(x_1, x_2)$, is in the form of,

  $$\dot{x} = \begin{bmatrix} 0 & a \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} b \\ c \end{bmatrix} u$$

  and this structure is always controllable for nonzero $a$, $b$ and $c$.

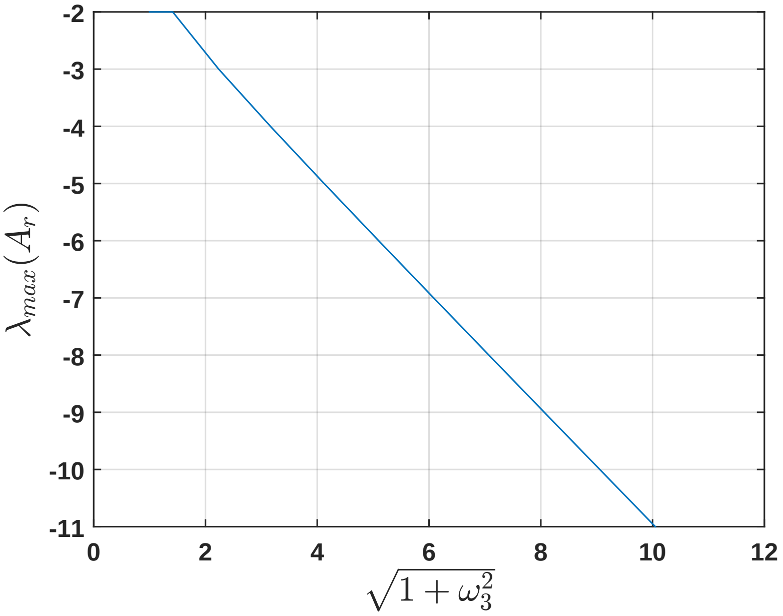


Figure 4.5: Maximum eigenvalue of $A_r$ for different yaw rates.

- Assumption *A3* of Theorem 2.4.1 is related to the controllability of $x_2$ dynamics. $x_2$ dynamics is a SISO system and it is controllable.

- Assumption *A4* of Theorem 2.4.1 is related to the bound on $\|f_1(0, x_2)\|$. As shown in (4.20), this bound exists.

- As given in (4.21), the Conditions *C1* and *C2* of Theorem 2.4.1 can be satisfied with the same bound value,

$$\gamma_1 = \gamma_2 = \sqrt{1+\omega_{3,max}^2}$$

### 4.5.3 Time dependent behaviour of the perturbation dynamics

Although norm constraints derived in the previous section can be used for selection of $A_r$, investigation of the time dependent behaviour of the perturbation equation reveals additional design considerations.

Consider the perturbation equation (4.19). With $\alpha = +1$, $B_r = \begin{bmatrix} 1 & 0 \end{bmatrix}$ and for diagonal $A_{rm}$ with eigenvalues $\lambda_{rm1}$ and $\lambda_{rm2}$ and similarly a diagonal $A_r$ with $\lambda_{r1}$ and $\lambda_{r2}$, it can be reformulated as,

$$\epsilon \dot{r} = \begin{bmatrix} -\lambda_{rm1} & p_2' \\ 0 & -\lambda_{rm2} \end{bmatrix} \begin{bmatrix} \delta \\ \omega_1 \end{bmatrix} + \begin{bmatrix} -p_1' + \lambda_{r1} & 0 \\ -a\omega_3 & \lambda_{r2} \end{bmatrix} r + \begin{bmatrix} 0 \\ d \end{bmatrix} \tag{4.22}$$

Equation (4.22) involves different dynamics working against each other. A stable reference model requires that $\lambda_{rm1\,,rm2} < 0$ which leads to positive definiteness for the first term of the perturbation dynamics. Effect of nonzero $(\delta(t), \omega_1(t))$ would be amplification of the signal. On the other hand, $(\delta(t), \omega_1(t))$ itself would decay in accordance with $A_{rm}$, due to reduced dynamics. By making $|\lambda_r| < |\lambda_{rm}|$, the dynamics of the perturbation can be adjusted so that perturbations do not respond to rapid changes in the uncontrolled states. This both smooths the response of the system and also decreases the convergence time.

Desired behaviour of the system, characterized by $A_{rm}$ is also bounded by the actuator bandwidth. With also the lower bound on $\|A_r\|$ for stability of the boundary layer equation, overall system should have a structure such that $|\lambda_r| < |\lambda_{rm}| < |\lambda_{actuator}|$.

The other term is the constant (or slowly varying) disturbance vector $\begin{bmatrix} 0 & d \end{bmatrix}^T$. The effect of constant disturbances is shifting of the origin of the system. Since the disturbance is in $w_1$ channel, it affect the perturbation on channel this channel only.

The presence of disturbance vector $d$ have other affects on the dynamics of the system and the equilibrium point of the overall system. These effects are discussed within the presentation of the numerical results in Section 4.6.

### 4.5.4 Overview of the design process

With the results of the analyses presented in this section, overall design process for the fault tolerant control of quadrotors is summarized below.

- Construct the fault detection part of the controller, using the techniques presented in Section 4.3.
- Renumber the rotor and adjust the $x - y$ coordinate system so that the faulty rotor becomes 4th propeller, in compliance with Figure 4.2.

- Determine the constant forces $f_{2,ftc}$ and $u_r = f_{max} - f_{2,ftc}$ that will be applied during the execution of the fault mitigation algorithm.
- Determine the maximum attainable yaw rate using (4.17) and choose $A_r$ according to Figure 4.5.
- Choose the reference model dynamics $A_{rm}$ such that eigenvalues of $A_{rm}$ satisfy,

  $$|\lambda_r| < |\lambda_{rm}| < |\lambda_{actuator}|$$

  where $\lambda_r$ and $\lambda_{actuator}$ represent the eigenvalues of $A_r$ and actuator dynamics respectively. 2-5 factor is suitable for selection of eigenvalues.
- Calculate the pitch rate disturbances ($q'$) according to

  $$\begin{aligned}\epsilon\dot{r} =& \alpha \cdot \left\{ \begin{bmatrix} 0 & p_2' \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \\ \omega_1 \end{bmatrix} + \begin{bmatrix} -p_1' \\ -a\omega_3 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} r + \begin{bmatrix} 0 \\ d \end{bmatrix} - A_{rm} \begin{bmatrix} \delta \\ \omega_1 \end{bmatrix} \right\} + A_r r \\ q' =& \begin{bmatrix} 1 & 0 \end{bmatrix} r\end{aligned}$$

  where $\delta$ is the angle between the current reduced attitude and the up direction, $\omega_1$ is the roll rate and $\omega_3$ is the yaw rate.
- Distance $\delta$ and reduced attitude parameters can be calculated with

  $$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = R_B^E \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} p_1' \\ p_2' \\ p_3' \end{bmatrix} = R_B^{E^T} \begin{bmatrix} \frac{-p_1 p_3}{\sqrt{p_1^2+p_2^2}} \\ \frac{-p_2 p_3}{\sqrt{p_1^2+p_2^2}} \\ \sqrt{p_1^2+p_2^2} \end{bmatrix}$$

- Apply the calculated pitch rate perturbation ($q'$) through the attitude rate controller of the system. Since only pitch channel will be controlled, a feedback linearized controller for pitch channel can be used for this purpose.

  $$\begin{aligned} u =& -a\omega_1\omega_3 + A_q\omega_2 + (q' + q_c) \\ M_c =& I_{yy}^T \cdot u = l \cdot (f_3 - f_1) \end{aligned}$$

  where $q_c$ is the output of the nominal controller. In accordance with (4.16), maximum input that can be applied is limited with,

  $$\max |u| \leq c f_{max}$$

The overall fault tolerant control system architecture is summarized in Figure 4.6. Numerical examples are presented in the next section.

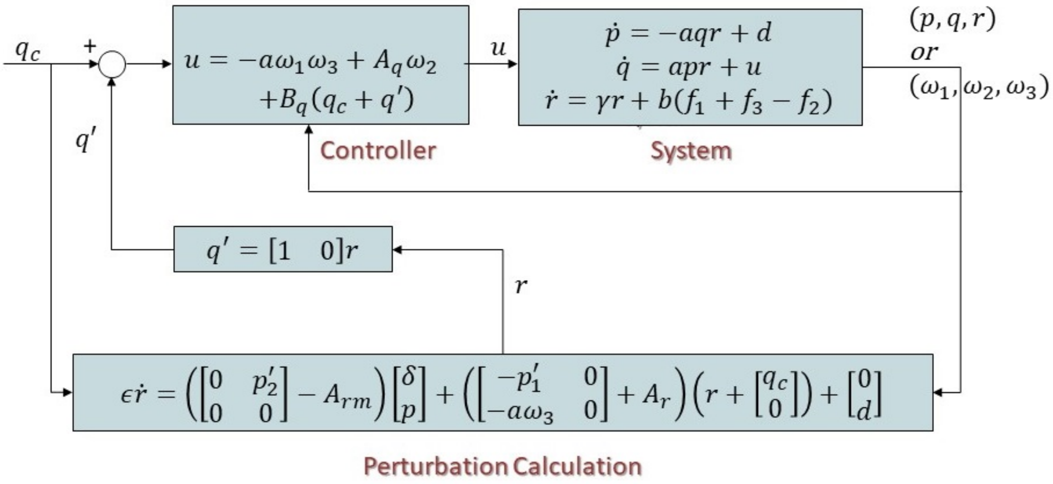

$q_c$
+
$q'$
$u = -a\omega_1\omega_3 + A_q\omega_2 + B_q(q_c + q')$
Controller
$u$
$\dot{p} = -aqr + d$
$\dot{q} = apr + u$
$\dot{r} = \gamma r + b(f_1 + f_3 - f_2)$
System
$(p, q, r)$
or
$(\omega_1, \omega_2, \omega_3)$
$q' = [1 \quad 0]r$
$r$
$\epsilon\dot{r} = \left(\begin{bmatrix}0 & p_2'\\0 & 0\end{bmatrix} - A_{rm}\right)\begin{bmatrix}\delta\\p\end{bmatrix} + \left(\begin{bmatrix}-p_1' & 0\\-a\omega_3 & 0\end{bmatrix} + A_r\right)\left(r + \begin{bmatrix}q_c\\0\end{bmatrix}\right) + \begin{bmatrix}0\\d\end{bmatrix}$
Perturbation Calculation

Figure 4.6: Fault tolerant control architecture for quadrotors

### 4.6 Attitude stabilization examples with reduced number of propellers

Mueller and D'Andrea provided system properties of the vehicles they have been working on. Numerical simulations are conducted with the quadrotor configuration whose parameters are taken from their work [106]. These parameters are shown in Table 4.2. With this set of parameters, auxiliary parameter $a$ can be calculated as,

$$a = \frac{\left(I_{zz}^T - I_{xx}^T\right)}{I_{xx}^T} = 0.9259$$

#### 4.6.1 Attitude Controller

Faulty conditions result in flight under high attitude angles and angular rates. Therefore a proportional plus derivative reduced attitude controller presented in Section 4.2 is chosen as in attitude controller. It's structure is repeated below for convenience.

$$\tau = k_p dist(p_0, q)\begin{bmatrix}-\langle Y_{p_0}^q, p_2\rangle\\ \langle Y_{p_0}^q, p_1\rangle\end{bmatrix} - K_d\begin{bmatrix}\omega_1\\ \omega_2\end{bmatrix} \tag{4.9}$$

The lower limit on $k_p$ for stability is,

$$k_p > \frac{\langle\omega, I\omega\rangle}{\pi^2 - dist\left(p_0, q\right)^2}$$

Table 4.2: Simulated system parameters

| **Property** | **Value** |
|---|---|
| Moment of Inertia Tensor: | $I^T_{xx} = I^T_{yy} = 2.7 \times 10^{-3}\,\mathrm{kgm}^2$<br>$I^T_{zz} = 5.2 \times 10^{-3}\,\mathrm{kgm}^2$ |
| Moment of Inertia of a Propeller: | $I^P_{zz} = 1.5 \times 10^{-5}\,\mathrm{kgm}^2$ |
| Mass: | $m = 0.5\,\mathrm{kg}$ |
| Rotor distance | $l = 0.17\,\mathrm{m}$ |
| Propeller force coefficient | $\kappa_f = 6.41 \times 10^{-6}\,\mathrm{Ns^2rad^{-2}}$ |
| Propeller torque coefficient | $\kappa_\tau = 1.72 \times 10^{-2}\,\mathrm{Nms^2rad^{-2}}$ |
| Propeller force limit | $f_{max} = 9.5\,\mathrm{N}$ |
| Motor time constant | $\sigma_M = 15\,\mathrm{ms}$ |
| Rotational drag coefficients | $\kappa_{dxx} = \kappa_{dyy} = 0.7 \times 10^{-5}\,\mathrm{Nms^2rad^2}$<br>$\kappa_{dzz} = 1.4 \times 10^{-4}\,\mathrm{Nms^2rad^2}$ |

For the system parameters given in Table 4.2, with the maximum angular error of $\pi/2$ $\mathrm{rad}$ and the maximum angular rate of $3\,\mathrm{rad\,s^{-1}}$, this limit can be calculated as $0.0129$. Following controller gain set is used in the simulations.

$$k_p = 0.05 \qquad K_d = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.02 \end{bmatrix}$$

For the feedback linearized part of the pitch rate controller, $A_q$ is chosen as $-20$.

### 4.6.2 Fault Tolerant Controller Parameters

Actuators are modeled as first order systems with time constant of $0.015\,\mathrm{ms}$. On the other hand, for maximum yaw rate of $3\,\mathrm{rad\,s^{-1}}$ results in minimum $A_r$ of $4.5$.

As explained in the previous section, reference model $A_{rm}$ and the perturbation control matrix $A_r$ should be chosen with an appropriate margin between them. Following set is used in simulations.

$$A_r = \begin{bmatrix} -10 & 0 \\ 0 & -10 \end{bmatrix} \qquad A_{rm} = \begin{bmatrix} -20 & 0 \\ 0 & -5 \end{bmatrix}$$

Small parameter $\epsilon$ is chosen as $\epsilon = 0.1$ and the sign parameter $\alpha$ is chosen as $\alpha = 1$.

Fault detection algorithm is constructed with the disturbance vector estimation using first three Chebyshev polynomials as the basis set. Weights are estimated with adaptation gain of $\Gamma = 10000$ and the following parameter set.

$$A_s = \begin{bmatrix} -50 & 0 & 0 \\ 0 & -50 & 0 \\ 0 & 0 & -50 \end{bmatrix} \qquad P_s = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$$

### 4.6.3 Simulation Results

Simulations are conducted for the failure of the 4th propeller at the $5\,\mathrm{s}$ of the flight. The plant is modeled as a continuous time system, while the flight control algorithm is run with discrete time step of $10\,\mathrm{ms}$

The system is started at the level position $(\phi, \theta, \psi) = (0, 0, 0)$ and trim is maintained at attitude $(\phi, \theta, \psi) = (15, 10, 0)$.

The fault is injected with loss of thrust of the 4th propeller. The rotation rate of the opposing propeller (2nd propeller) is set to the predetermined value, immediately after the detection of the fault and fault tolerant controller is activated. Calculated disturbance vectors are shown in Figure 4.7, where it can be seen that it took about $4\,\mathrm{s}$ for initial convergence of the gains.

The moment of an unbalanced rotor is very high. Therefore faulty condition can be identified rather rapidly . It takes $30\,\mathrm{ms}$ for the FDI algorithm to detect the fault (Figure 4.8). After that, it takes approximately $70\,\mathrm{ms}$ for the 2nd propeller to slow down to the desired level. This situation is shown in Figure 4.9, for the case where $f_{2,ftc} = 0.2\,\mathrm{N}$.

In order to demonstrate the effect of the thrust level on the opposing propeller, simulations are conducted for $f_{2,ftc}$ values of $0$, $0.1$ and $0.2\,\mathrm{N}$, where $0\,\mathrm{N}$ corresponds to the flight with two propellers case.

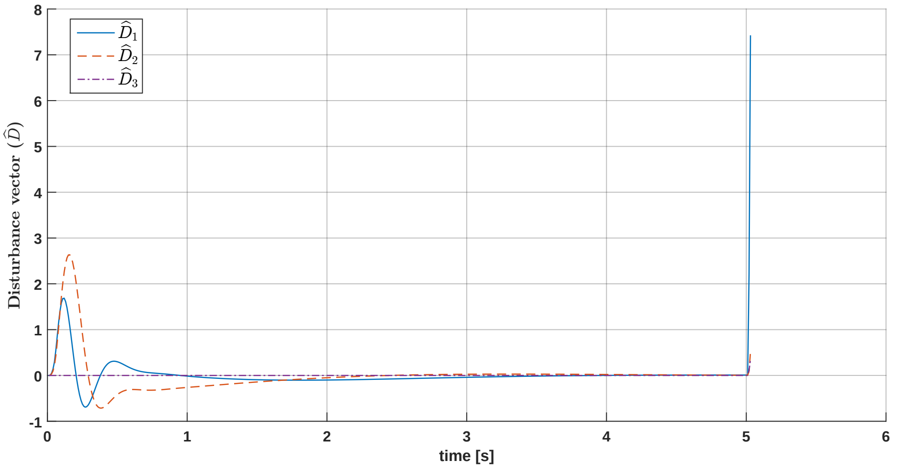


Figure 4.7: Calculated disturbance vector for the quadrotor problem

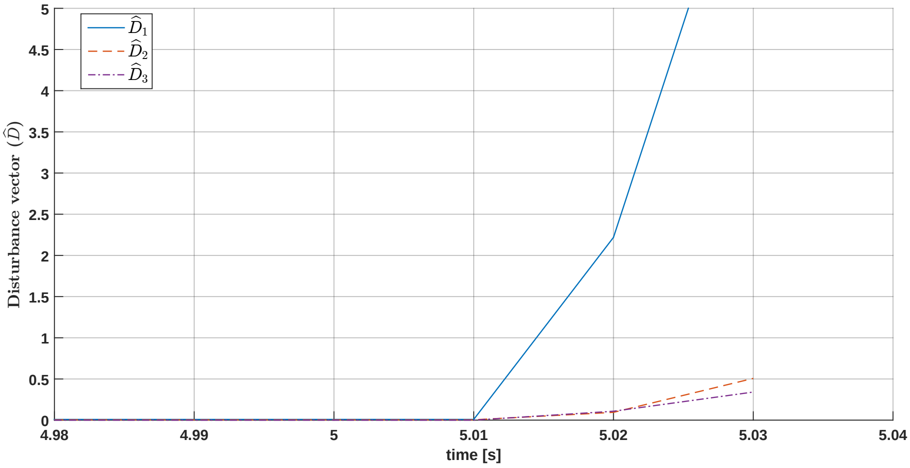


Figure 4.8: Elements of the disturbance vector during the occurrence of the fault

In order to the evaluate the performance of the control system, angle between the positive $z$ axis of the propeller (Figure 4.2) and the local level plane is compared in 4.10. The two-propeller case converges to the up position with a small offset of few degrees. The effect of second propeller can be seen with the increasing values of $f_{2,ftc}$. This thrust tilts the equilibrium attitude of the rotation vector with a magnitude proportional to the applied thrust. In compliance with the theoretical discussions, pitch rate perturbations converge to zero for $f_{2,ftc} = 0$ but presence of a disturbance moment ($d$) shifts the equilibrium point of the perturbation (Figure 4.12).

While the pitch angle remains mainly constant, it rotates around the local vertical axis

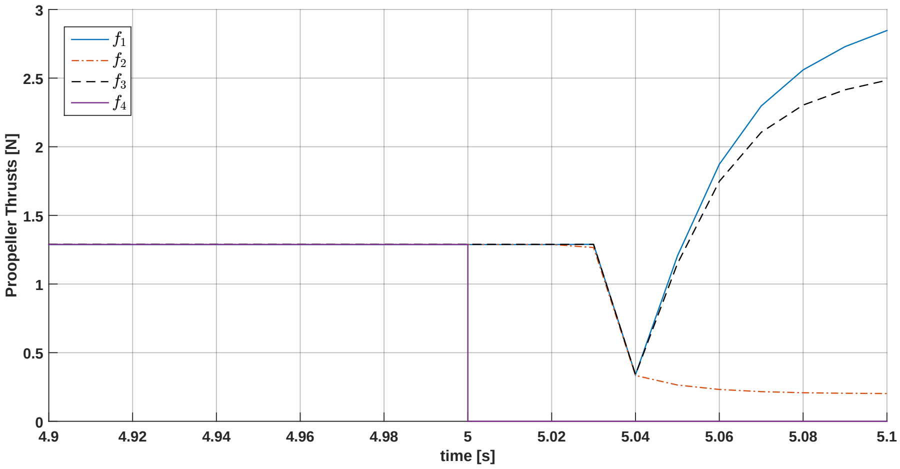


Figure 4.9: Applied forces on the propellers, during the occurrence of the fault

on the $x - y$ plane. This angle is shown in Figure 4.11. The resultant motion is similar to a spinning top and it can be visualized as a free spinning top with rotation axis is tilted.

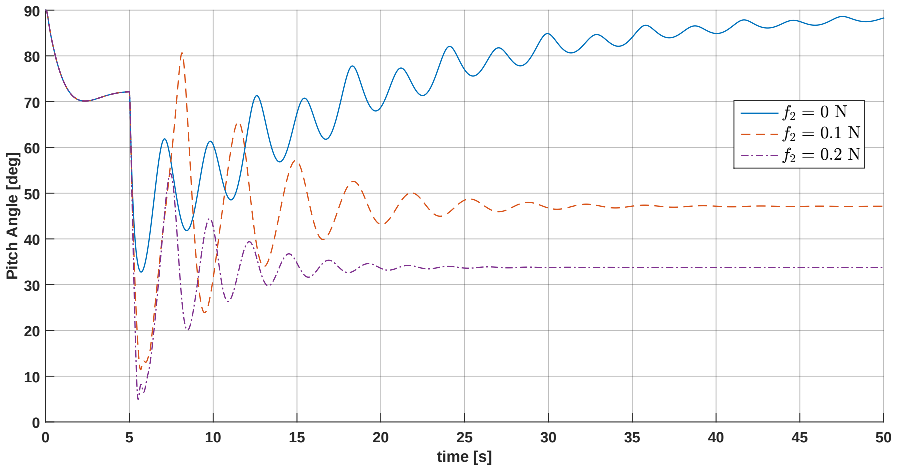


Figure 4.10: Angle between the positive $z$ axis of the propeller and the local level plane

For completeness; Euler angles, attitude angular rates and applied forces on the propellers for $f_{2,ftc} = 0$ and $f_{2,ftc} = 0.2$ cases are shown in Figures 4.13, 4.14, 4.15.

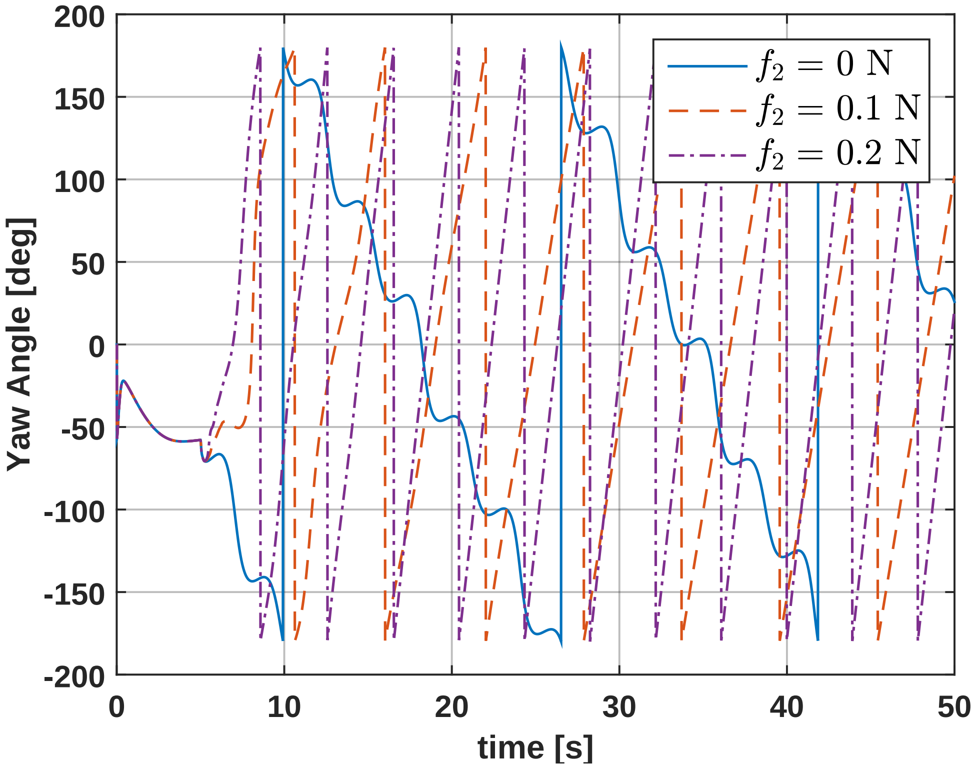


Figure 4.11: Yaw angle of the attitude vector on the $x - y$ local level plane

## 4.7 Summary

In this chapter, application of the proposed fault tolerant control structure on the quadrotor attitude control problem is explained. The discussion is limited to the loss of single propeller case, since it involves all the essentials of the proposed FTC implementation. Loss of two co-rotating propeller case is also included within the proposed structure. Similar analysis can be conducted for loss of two anti-rotating propellers or flight with single propeller cases but they are omitted for brevity.

Attitude control of a spinning body is a complicated problem. But the simulation results indicate that proposed controller structure is can be effectively used in such systems. The proposed methodology can also be extended to other systems which uses different mechanisms for control, as well.

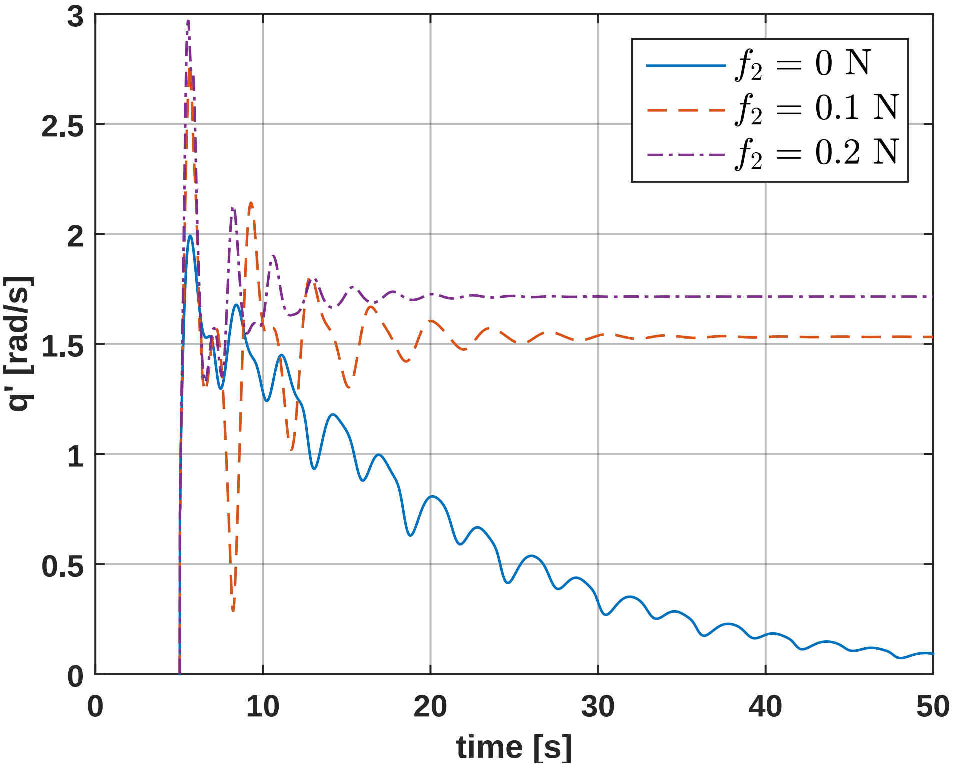
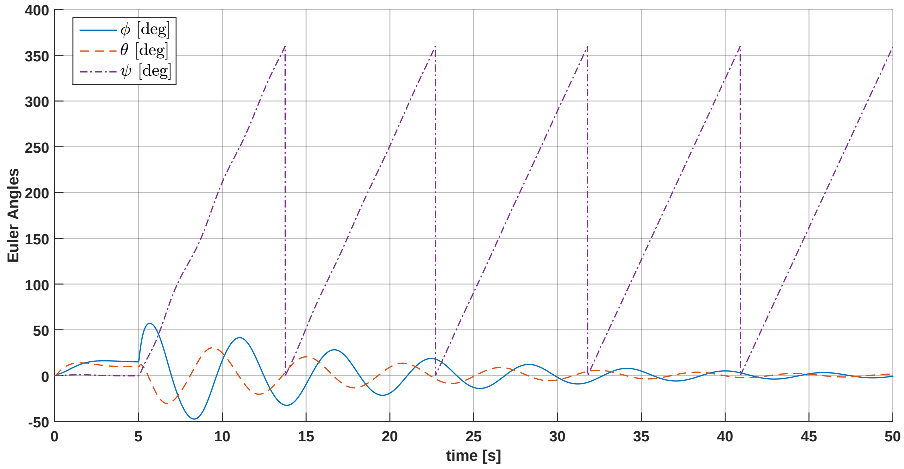
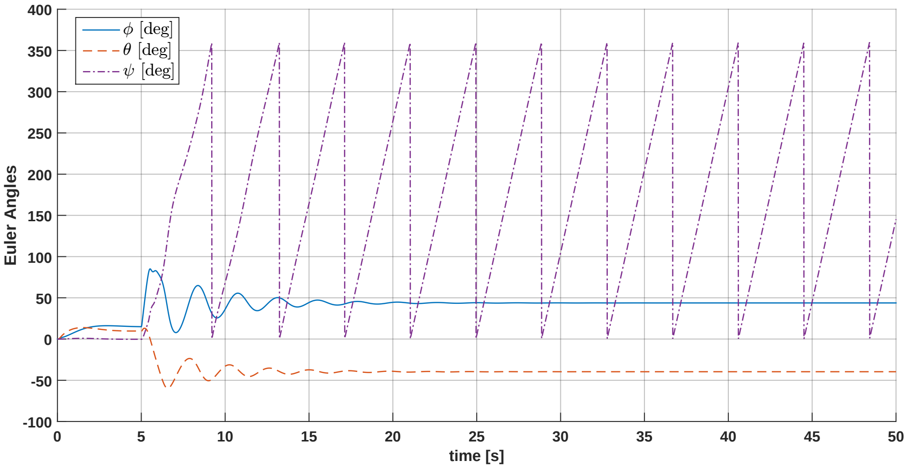


Figure 4.12: Pitch rate perturbations

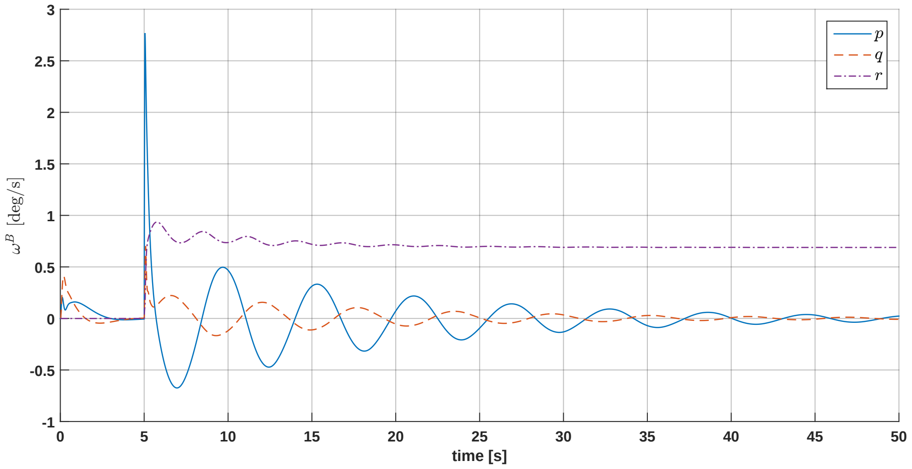


(a) $f_{2,ftc} = 0$

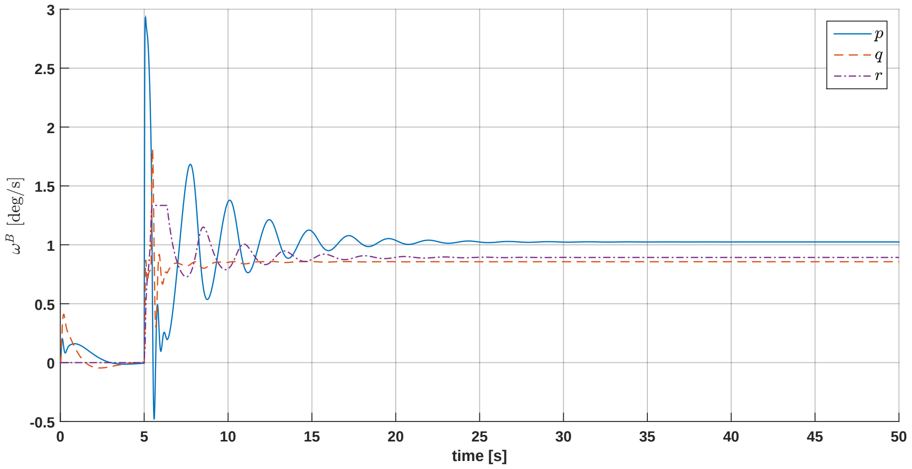


(b) $f_{2,ftc} = 0.2$

Figure 4.13: Euler angles of the quadrotor for different applied $f_{2,ftc}$ values

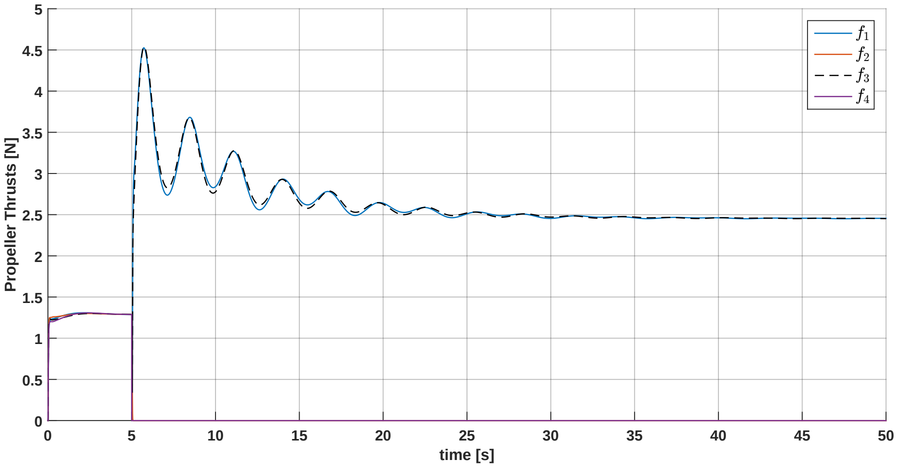


(a) $f_{2,ftc} = 0$

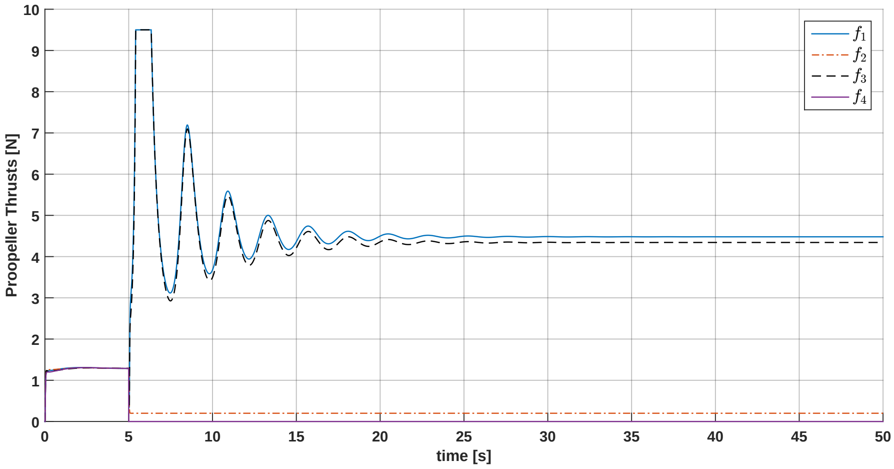


(b) $f_{2,ftc} = 0.2$

Figure 4.14: Angular rates of the quadrotor for different applied $f_{2,ftc}$ values



(a) $f_{2,ftc} = 0$



(b) $f_{2,ftc} = 0.2$

Figure 4.15: Applied propeller thrusts for different $f_{2,ftc}$ values

# CHAPTER 5

# CONCLUSION

The starting point of this thesis work was the observation that in some nonlinear systems, it is possible to find special control laws that would stabilize an unstable internal state without directly controlling it. However, such solutions are highly system specific and therefore the aim of this research was to find a systematic way to construct control systems that would achieve this for a large class of systems, so that it can be used as an algorithmic fault tolerance measure.

This is achieved with a novel adaptive control structure that generates perturbations on the reference signal of the directly actuated states, in order to stabilize the unactuated internal states of the system. Theoretical development of this control architecture is presented in Chapter 2, together with a theorem that includes the assumptions, conditions and results of the proposed method and the original contributions of the thesis work is explained in Chapter 1 Section 1.5. As a concluding remark, main achievements and possible extensions can be summarized as follows.

The proposed control system architecture includes an external dynamic system formulation, which generates perturbations on the main trajectory of the controlled states that stabilizes the internal dynamics (uncontrolled states) of the system. This is especially useful if the internal dynamics of the system is unstable, *i.e.*, non-minimum phase. As explained in Section 1.2 through a literature review on the subject, this is a very important property and the developed method can be systematically applied to different problems. However, following comments can be made for application of the proposed control system and possible research directions.

The perturbation generator is formulated as a singularly perturbed dynamic system and results of the Tikhonov's theorem (Theorem 2.3.1) is used. The main statement of the Tikhonov's theorem is an asymptotic order of magnitude comparison on the convergence of the singularly perturbed system to the solution of the unperturbed one. Using this result, it is proved that the uncontrolled states are bounded. However, it is not possible to calculate the exact value of this bound, only it is known that it exists and as the perturbation parameter goes to zero ($\epsilon \to 0$), this bound converges the bound of the reference model. Therefore, in practical applications, maximum attainable values should be checked through simulations.

The design methodology is mainly derived from the stability analysis of the resultant boundary layer equation. This analysis is conducted through stability analysis of the linearized system. Therefore validity of this assumption should be checked, especially for large perturbation values. On the other hand, the theories on stability of perturbed systems is highly used for the development of the method. This provided -probably conservative- bounds on the attainable control systems. It may also be possible to find more suitable perturbation generators to construct exponentially stable boundary layer equations for a more specific system form.

Although the developed line of analysis provides the conditions for asymptotic convergence of the cascade system to a reference model, it does not provide information on the rate of converge. This is an important question and unfortunately, it is beyond my mathematical background to develop necessary theoretical work to answer that question.

Apart from above mentioned possible further theoretical research directions, following comments can be made for practical applications.

Lagrangian form of equation of motion is widely used especially in robotic control problems. Application on such class of systems are shown in Chapter 3. Although numerical applications are provided for limited number of cases, the results and methodology are applicable to all Lagrangian systems.

Underactuated robotics is a very important class of robotic systems, where motion of a robot is achieved with less number of actuators than the degrees of freedom of the system. This is also related to achieve robot motion similar to humans and animals

(Locomotion). Perturbation generation mechanism developed in this thesis can also be seen as a tool for achieving such kind of motion. Application in this direction would be an important improvement over present, more system specific, solutions.

Attitude stabilization with reduced number of actuators is another interesting problem that have been addressed. This problem is addressed with quadrotor example in Chapter 4. The proposed method not only provides a fault tolerance measure for multirotor UAVs, but also gives way to novel vehicle configurations. A possible extension to this work would be control of insect-like vehicles with high frequency flapping motion modeled as a perturbation signal.

As a final remark, it should be noted that coupled nonlinear systems may have more than one possible state representations. For example in underactuated robotics, the equations of motion can be written equivalently in collocated or non-collocated forms (see Section 3.1.2.2 in Chapter 3). Particle choice may result in different formulation of perturbation dynamics and therefore it may be possible to achieve better results for the example applications, than the ones presented in this thesis.

# BIBLIOGRAPHY

[1] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, and J. Schröder, *Diagnosis and fault-tolerant control*. Springer, 2006, vol. 2.

[2] R. Patton, "Fault detection and diagnosis in aerospace systems using analytical redundancy", *Computing & Control Engineering Journal*, vol. 2, no. 3, pp. 127–136, 1991.

[3] *The average cost of iot sensors is falling*, Available at `https://www.theatlas.com/charts/BJsmCFAl` [17/08/2018].

[4] H. K. Khalil, *Nonlinear systems*, 2nd ed. Prentice Hall New Jersey, 1996.

[5] C. I. Byrnes, A. Isidori, and J. C. Willems, "Passivity, feedback equivalence, and the global stabilization of minimum phase nonlinear systems", *IEEE Transactions on Automatic Control*, vol. 36, no. 11, pp. 1228–1240, 1991.

[6] L. Marconi, L. Praly, and A. Isidori, "Output stabilization via nonlinear Luenberger observers", *SIAM Journal on Control and Optimization*, vol. 45, no. 6, pp. 2277–2298, 2007.

[7] L. Marconi and L. Praly, "Uniform practical nonlinear output regulation", *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1184–1202, 2008.

[8] S. Nazrulla and H. K. Khalil, "Robust stabilization of non-minimum phase nonlinear systems using extended high-gain observers", *IEEE Transactions on Automatic Control*, vol. 56, no. 4, pp. 802–813, 2011.

[9] M.-L. Chiang and A. Isidori, "Nonlinear output regulation with saturated control for a class of non-minimum phase systems", in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, IEEE, 2015, pp. 7677–7682.

[10] I. Kanellakopoulos, P. V. Kokotovic, and A. S. Morse, "Systematic design of adaptive controllers for feedback linearizable systems", in *American Control Conference, 1991*, IEEE, 1991, pp. 649–654.

[11] M. Jankovic, R. Sepulchre, and P. V. Kokotovic, “Constructive Lyapunov stabilization of nonlinear cascade systems”, *IEEE Transactions on Automatic Control*, vol. 41, no. 12, pp. 1723–1735, 1996.

[12] W. Su and M. Fu, “Robust nonlinear control: Beyond backstepping and nonlinear forwarding”, in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, IEEE, vol. 1, 1999, pp. 831–836.

[13] D. Swaroop, J. K. Hedrick, P. P. Yip, and J. C. Gerdes, “Dynamic surface control for a class of nonlinear systems”, *IEEE Transactions on Automatic Control*, vol. 45, no. 10, pp. 1893–1899, 2000.

[14] C. S. Teodorescu, H. Siguerdidjane, S. Olaru, and A. Arzandé, “Towards robust control with constraints for a class of dynamical systems: Theory and application”, in *American Control Conference (ACC), 2010*, IEEE, 2010, pp. 4701–4706.

[15] J.-J. E. Slotine and W. Li, *Applied nonlinear control*, 1. Prentice-Hall Englewood Cliffs, NJ, 1991, vol. 199.

[16] A. Narang-Siddarth and J. Valasek, “A constructive stabilization approach for open-loop unstable non-affine systems”, in *American Control Conference (ACC), 2013*, IEEE, 2013, pp. 5685–5689.

[17] D. Ho and J. K. Hedrick, “Control of nonlinear non-minimum phase systems with input-output linearization”, in *American Control Conference (ACC), 2015*, IEEE, 2015, pp. 4016–4023.

[18] S. R. Nekoo and B. Geranmehr, “Control of non-affine systems using the state-dependent riccati equation (sdre)”, *Majlesi Journal of Mechatronic Systems*, vol. 2, no. 4, 2013.

[19] C. Sun, B. Jing, and Z. Liu, “Adaptive neural control of nonaffine nonlinear systems without differential condition for nonaffine function”, *Mathematical Problems in Engineering*, vol. 2016, 2016.

[20] C. Shi, X. Dong, J. Xue, Y. Chen, and J. Zhi, “Robust adaptive neural control for a class of non-affine nonlinear systems”, *Neurocomputing*, vol. 223, pp. 118–128, 2017.

[21] Y. Zhang, T. Chai, and D. Wang, “An alternating identification algorithm for a class of nonlinear dynamical systems”, *IEEE Transactions on Neural Networks and Learning Systems*, 2016.

[22] L.-B. Wu and G.-H. Yang, “Adaptive fault-tolerant control of a class of nonaffine nonlinear systems with mismatched parameter uncertainties and disturbances”, *Nonlinear Dynamics*, vol. 82, no. 3, pp. 1281–1291, 2015.

[23] O. Pettersson, “Execution monitoring in robotics: A survey”, *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 73–88, 2005.

[24] J. Gertler, *Fault detection and diagnosis*. Springer, 2015.

[25] R. Isermann, *Fault-diagnosis systems: An introduction from fault detection to fault tolerance*. Springer Science and Business Media, 2006.

[26] ——, “Model-based fault-detection and diagnosis – Status and applications”, *Annual Reviews in control*, vol. 29, no. 1, pp. 71–85, 2005.

[27] I. Hwang, S. Kim, Y. Kim, and C. E. Seah, “A survey of fault detection, isolation, and reconfiguration methods”, *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 636–653, May 2010.

[28] R. J. Patton, “Fault-tolerant control”, *Encyclopedia of Systems and Control*, pp. 422–428, 2015.

[29] Z. Gao, C. Cecati, and S. X. Ding, “A survey of fault diagnosis and fault-tolerant techniques - Part I: Fault diagnosis with model-based and signal-based approaches”, *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[30] Y. Zhang and J. Jiang, “Bibliographical review on reconfigurable fault-tolerant control systems”, *Annual reviews in control*, vol. 32, no. 2, pp. 229–252, 2008.

[31] F. Bateman, H. Noura, and M. Ouladsine, “An actuator fault detection, isolation and estimation system for an uav using input observers”, in *Control Conference (ECC), 2007 European*, IEEE, 2007, pp. 1805–1810.

[32] M. L. McIntyre, W. E. Dixon, D. M. Dawson, and I. D. Walker, “Fault identification for robot manipulators”, *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 1028–1034, Oct. 2005.

[33] G. Ducard, “Actuator fault detection in uavs”, in *Handbook of Unmanned Aerial Vehicles*, Springer, 2015, pp. 1071–1122.

[34] L. Wu, W. Luo, Y. Zeng, F. Li, and Z. Zheng, “Fault detection for underactuated manipulators modeled by markovian jump systems”, *IEEE Transactions on Industrial Electronics*, vol. 63, no. 7, pp. 4387–4399, 2016.

[35] B. D. O. Anderson, “Failures of adaptive control theory and their resolution”, *Communications in Information & Systems*, vol. 5, no. 1, pp. 1–20, 2005.

[36] T. A. Johansen and T. I. Fossen, “Control allocation—a survey”, *Automatica*, vol. 49, no. 5, pp. 1087–1103, 2013.

[37] R. C. Hoover, R. G. Roberts, A. A. Maciejewski, P. S. Naik, and K. M. Ben-Gharbia, “Designing a failure-tolerant workspace for kinematically redundant robots”, *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1421–1432, 2015.

[38] N. Hovakimyan and C. Cao, $\mathcal{L}_1$ *Adaptive control theory: Guaranteed robustness with fast adaptation*. SIAM, 2010.

[39] P. A. Ioannou, A. M. Annaswamy, K. S. Narendra, S. Jafari, L. Rudd, R. Ortega, and J. Boskovic, “$\mathcal{L}_1$ - adaptive control: Stability, robustness, and interpretations”, *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 3075–3080, 2014.

[40] R. Ortega and E. Panteley, “Adaptation is unnecessary in $\mathcal{L}_1$ - “adaptive” control”, *IEEE Control Systems*, vol. 36, no. 1, pp. 47–52, 2016.

[41] G. Chowdhary and E. Johnson, “Concurrent learning for convergence in adaptive control without persistency of excitation”, in *Decision and Control (CDC), 2010 49th IEEE Conference on*, IEEE, 2010, pp. 3674–3679.

[42] N. Nguyen, J. Burken, and A. Ishihara, “Least-squares adaptive control using chebyshev orthogonal polynomials”, in *AIAA Infotech@ Aerospace Conference, AIAA-2011-1402*, 2011.

[43] I. Kolmanovsky, E. Garone, and S. Di Cairano, “Reference and command governors: A tutorial on their theory and automotive applications”, in *American Control Conference (ACC), 2014*, IEEE, 2014, pp. 226–241.

[44] T. Yucelen and E. Johnson, “A new command governor architecture for transient response shaping”, *International Journal of Adaptive Control and Signal Processing*, vol. 27, no. 12, pp. 1065–1085, 2013.

[45] G. De La Torre, T. Yucelen, and E. N. Johnson, “A new model reference control architecture: Stability, performance, and robustness”, *International Journal of Robust and Nonlinear Control*, vol. 26, no. 11, pp. 2355–2377, 2016.

[46] T. Yucelen and E. Johnson, “Command governor-based adaptive control”, in *AIAA Guidance, Navigation, and Control Conference*, 2012, p. 4618.

[47] H. J. Sussmann and W. Liu, “Limits of highly oscillatory controls and the approximation of general paths by admissible trajectories”, in *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on*, IEEE, 1991, pp. 437–442.

[48] R. M. Murray and S. S. Sastry, “Nonholonomic motion planning: Steering using sinusoids”, *IEEE Transactions on Automatic Control*, vol. 38, no. 5, pp. 700–716, 1993.

[49] L. Schenato, “Analysis and control of flapping flight: From biological to robotic insects”, PhD thesis, University of California, Berkeley, 2003.

[50] P. V. Kokotovic, “Applications of singular perturbation techniques to control problems”, *SIAM Review*, vol. 26, no. 4, pp. 501–550, 1984.

[51] N. Hovakimyan, E. Lavretsky, and A. Sasane, “Dynamic inversion for nonaffine-in-control systems via time-scale separation. part I”, *Journal of Dynamical and Control Systems*, vol. 13, no. 4, pp. 451–465, 2007.

[52] E. Lavretsky and N. Hovakimyan, “Adaptive dynamic inversion for nonaffine-in-control uncertain systems via time-scale separation. part II”, *Journal of Dynamical and Control Systems*, vol. 14, no. 1, pp. 33–41, 2008.

[53] N. Hovakimyan, E. Lavretsky, and C. Cao, “Adaptive dynamic inversion via time-scale separation”, *IEEE Transactions on Neural Networks*, vol. 19, no. 10, pp. 1702–1711, Oct. 2008.

[54] J. Teo and J. P. How, “Equivalence between approximate dynamic inversion and proportional-integral control”, in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, IEEE, 2008, pp. 2179–2183.

[55] J. Teo, J. P. How, and E. Lavretsky, “Proportional-integral controllers for minimum-phase nonaffine-in-control systems”, *IEEE Transactions on Automatic Control*, vol. 55, no. 6, pp. 1477–1483, 2010.

[56] ——, “On approximate dynamic inversion and proportional-integral control”, in *2009 American Control Conference*, IEEE, 2009, pp. 1592–1597.

[57] V. D. Yurkevich, “Output regulation of pulse-width-modulated nonlinear nonaffine-in-control systems via singular perturbation”, *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1374–1379, 2011.

[58] P. Krishnamurthy and F. Khorrami, “A singular perturbation based approach for systems with nonlinear input uncertainties”, in *Control and Decision Conference (CCDC), 2011 Chinese*, IEEE, 2011, pp. 1477–1482.

[59] A. Narang-Siddarth and J. Valasek, *Nonlinear time scale systems in standard and nonstandard forms: Analysis and control*. SIAM, 2014.

[60] H. K. Khalil, *Lecture notes: Nonlinear control*, Available at `http://www.egr.msu.edu/~khalil/NonlinearControl/` [23/10/2017].

[61] ——, *Nonlinear control*, 1st ed. Pearson New York, 2015.

[62] R. W. Brockett, “Asymptotic stability and feedback stabilization”, *Differential Geometric Control Theory*, vol. 27, no. 1, pp. 181–191, 1983.

[63] D. Luenberger, “An introduction to observers”, *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 596–602, 1971.

[64] G. Ciccarella, M. Dalla Mora, and A. Germani, “A Luenberger-like observer for nonlinear systems”, *International Journal of Control*, vol. 57, no. 3, pp. 537–556, 1993.

[65] C. Kravaris, J. Hahn, and Y. Chu, “Advances and selected recent developments in state and parameter estimation”, *Computers & Chemical Engineering*, vol. 51, pp. 111–123, 2013.

[66] C. Kravaris, V. Sotiropoulos, C. Georgiou, N. Kazantzis, M. Xiao, and A. J. Krener, “Nonlinear observer design for state and disturbance estimation”, *Systems & Control Letters*, vol. 56, no. 11, pp. 730–735, 2007.

[67] R. M. Sanner and J.-J. Slotine, “Gaussian networks for direct adaptive control”, *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 837–863, 1992.

[68] J.-B. Pomet and L. Praly, “Adaptive nonlinear regulation: Estimation from the Lyapunov equation”, *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 729–740, 1992.

[69] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, “Robotic fault detection and fault tolerance: A survey”, *Reliability Engineering & System Safety*, vol. 46, no. 2, pp. 139–158, 1994.

[70] ——, “A dynamic fault tolerance framework for remote robots”, *IEEE Transactions on Robotics and Automation*, vol. 11, no. 4, pp. 477–490, 1995.

[71] W. E. Dixon, I. D. Walker, D. M. Dawson, and J. P. Hartranft, “Fault detection for robot manipulators with parametric uncertainty: A prediction-error-based approach”, *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 689–699, 2000.

[72] A. T. Vemuri and M. M. Polycarpou, “A methodology for fault diagnosis in robotic systems using neural networks”, *Robotica*, vol. 22, no. 04, pp. 419–438, 2004.

[73] F. Caccavale, A. Marino, G. Muscio, and F. Pierri, “Discrete-time framework for fault diagnosis in robotic manipulators”, *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1858–1873, 2013.

[74] I. Eski, S. Erkaya, S. Savas, and S. Yildirim, “Fault detection on robot manipulators using artificial neural networks”, *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 1, pp. 115–123, 2011.

[75] Y. Cheng, R. Wang, and M. Xu, “A combined model-based and intelligent method for small fault detection and isolation of actuators”, *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2403–2413, 2016.

[76] B. T. Thumati, G. R. Halligan, and S. Jagannathan, “A novel fault diagnostics and prediction scheme using a nonlinear observer with artificial immune system as an online approximator.”, *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 569–578, 2013.

[77] T. Yüksel and A. Sezgin, “A model-based fault function approximation scheme for robot manipulators using m-anfis”, in *MELECON 2010- 15th IEEE Mediterranean Electrotechnical Conference*, IEEE, 2010, pp. 47–51.

[78] W.-H. Chen, D. J. Ballance, P. J. Gawthrop, and J. O’Reilly, “A nonlinear disturbance observer for robotic manipulators”, *IEEE Transactions on Industrial Electronics*, vol. 47, no. 4, pp. 932–938, 2000.

[79] D. Brambilla, L. M. Capisani, A. Ferrara, and P. Pisu, “Fault detection for robot manipulators via second-order sliding modes”, *IEEE Transactions on Industrial Electronics*, vol. 55, no. 11, pp. 3954–3963, 2008.

[80] L. M. Capisani, A. Ferrara, A. F. de Loza, and L. M. Fridman, “Manipulator fault diagnosis via higher order sliding-mode observers”, *IEEE Transactions on Industrial Electronics*, vol. 59, no. 10, pp. 3979–3986, 2012.

[81] M. Van, P. Franciosa, and D. Ceglarek, “Fault diagnosis and fault-tolerant control of uncertain robot manipulators using high-order sliding mode”, *Mathematical Problems in Engineering*, vol. 2016, 2016.

[82] B. Xiao, S. Yin, and H. Gao, “Reconfigurable tolerant control of uncertain mechanical systems with actuator faults: A sliding mode observer-based approach”, *IEEE Transactions on Control Systems Technology*, 2017.

[83] B. Xiao and S. Yin, “An intelligent actuator fault reconstruction scheme for robotic manipulators”, *IEEE Transactions on Cybernetics*, vol. 48, no. 2, pp. 639–647, 2018.

[84] A. Mohammadi, H. J. Marquez, and M. Tavakoli, “Nonlinear disturbance observers: Design and applications to Euler-Lagrange systems”, *IEEE Control Systems*, vol. 37, no. 4, pp. 50–72, 2017.

[85] Y. Oh and W. K. Chung, “Disturbance-observer-based motion control of redundant manipulators using inertially decoupled dynamics”, *IEEE/ASME Transactions on Mechatronics*, vol. 4, no. 2, pp. 133–146, 1999.

[86] M. J. Kim, Y. J. Park, and W. K. Chung, "Design of a momentum-based disturbance observer for rigid and flexible joint robots", *Intelligent Service Robotics*, vol. 8, no. 1, pp. 57–65, 2015.

[87] H. Chang, P. Huang, M. Wang, and Z. Lu, "Locked-joint failure identification for free-floating space robots", in *Information and Automation (ICIA), 2014 IEEE International Conference on*, IEEE, 2014, pp. 170–175.

[88] C. L. Lewis and A. A. Maciejewski, "Fault tolerant operation of kinematically redundant manipulators for locked joint failures", *IEEE Transactions on Robotics and Automation*, vol. 13, no. 4, pp. 622–629, 1997.

[89] M. Goel, A. A. Maciejewski, and V. Balakrishnan, "Analyzing unidentified locked-joint failures in kinematically redundant manipulators", *Journal of Field Robotics*, vol. 22, no. 1, pp. 15–29, 2005.

[90] C. Xianbao, G. Feng, Q. Chenkun, and T. Xinghua, "Gait planning for a quadruped robot with one faulty actuator", *Chinese Journal of Mechanical Engineering*, vol. 28, no. 01, p. 1, 2015.

[91] Z. Mu, B. Zhang, W. Xu, B. Li, and B. Liang, "Fault tolerance kinematics and trajectory planning of a 6-dof space manipulator under a single joint failure", in *Real-time Computing and Robotics (RCAR), IEEE International Conference on*, IEEE, 2016, pp. 483–488.

[92] H. Du and F. Gao, "Fault tolerance properties and motion planning of a six-legged robot with multiple faults", *Robotica*, pp. 1–18, 2016.

[93] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[94] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. Wiley New York, 2006, vol. 3.

[95] R. Tedrake, *Underactuated robotics: algorithms for walking, running, swimming, flying, and manipulation (course notes for mit 6.832)*, Available at `http://underactuated.mit.edu/underactuated.html` [28/11/2017].

[96] H. Wang and S. Yu, "Tracking control of robot manipulators via orthogonal polynomials neural network", *Advances in Neural Networks–ISNN 2009*, pp. 178–187, 2009.

[97] M. W. Spong, "Partial feedback linearization of underactuated mechanical systems", in *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on*, vol. 1, Sep. 1994, 314–321 vol.1.

[98] L. Techy, C. K. Reddy, C. A. Woolsey, C. Cao, and N. Hovakimyan, "Nonlinear control of a novel two-link pendulum", in *2007 American Control Conference*, Jul. 2007, pp. 19–24.

[99] J. She, A. Zhang, X. Lai, and M. Wu, "Global stabilization of 2-dof underactuated mechanical systems—an equivalent-input-disturbance approach", *Nonlinear Dynamics*, vol. 69, no. 1, pp. 495–509, 2012.

[100] S. Oh and K. Kong, "Two-degree-of-freedom control of a two-link manipulator in the rotating coordinate system", *IEEE Transactions on Industrial Electronics*, vol. 62, no. 9, pp. 5598–5607, 2015.

[101] J. Wilson, M. Charest, and R. Dubay, "Non-linear model predictive control schemes with application on a 2 link vertical robot manipulator", *Robotics and Computer-Integrated Manufacturing*, vol. 41, pp. 23–30, 2016.

[102] K. Dolinsky and S. Celikovsky, "Adaptive nonlinear tracking for robotic walking", *Cybernetics and Physics*, vol. 1, no. 1, pp. 28–35, 2012.

[103] D. Pucci, F. Romano, and F. Nori, "Collocated adaptive control of underactuated mechanical systems", *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1527–1536, 2015.

[104] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints", *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 620–629, 2016.

[105] X.-Z. Lai, J.-H. She, S. X. Yang, and M. Wu, "Comprehensive unified control strategy for underactuated two-link manipulators", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 389–398, 2009.

[106] M. W. Mueller and R. D'Andrea, "Relaxed hover solutions for multicopters: Application to algorithmic redundancy and novel vehicles", *The International Journal of Robotics Research*, vol. 35, no. 8, pp. 873–889, 2016.

[107] M. W. Mueller and R. D'Andrea, "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers", in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, IEEE, 2014, pp. 45–52.

[108] A. Freddi, A. Lanzon, and S. Longhi, "A feedback linearization approach to fault tolerance in quadrotor vehicles", *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 5413–5418, 2011.

[109] A. Lanzon, A. Freddi, and S. Longhi, "Flight control of a quadrotor vehicle subsequent to a rotor failure", *Journal of Guidance, Control, and Dynamics*, 2014.

[110] A. Akhtar, S. L. Waslander, and C. Nielsen, "Fault tolerant path following for a quadrotor", in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, IEEE, 2013, pp. 847–852.

[111] A. Simha, S. Vadgama, and S. Raha, "A geometric approach to rotor failure tolerant trajectory tracking control design for a quadrotor", arXiv preprint arXiv:1704.00327, 2017.

[112] G. Michieletto, M. Ryll, and A. Franchi, "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*, 2017, pp. 2747–2752.

[113] M. Saied, H. Shraim, C. Francis, I. Fantoni, and B. Lussier, "Controllability analysis and motors failures symmetry in a coaxial octorotor", in *Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), 2015 Third International Conference on*, IEEE, 2015, pp. 245–250.

[114] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor", *IEEE Robotics & Automation Magazine*, vol. 19, pp. 20–32, 2012.

[115] C. I. Byrnes and A. Isidori, "On the attitude stabilization of rigid spacecraft", *Automatica*, vol. 27, no. 1, pp. 87–95, 1991.

[116] F. Bullo, R. M. Murray, and A. Sarti, "Control on the sphere and reduced attitude stabilization", California Institute of Technology, 1995.

[117] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch, “Rigid-body attitude control”, *IEEE Control Systems*, vol. 31, no. 3, pp. 30–51, 2011.

[118] L. W. Kheng, *Matrix differentiation - CS5240 CSheoretical foundations in multimedia*, Available at `www.comp.nus.edu.sg/~cs5240/lecture/matrix-differentiation.pdf` [28/11/2017].

# APPENDIX A

# MATRIX DIFFERENTIATION RULES

Matrix differentiation is used for derivation of some of the formulations. These differentiation rules are summarized in this appendix.

Matrix differentiation formulations can be found in many references, therefore proofs are omitted. Particular source for the preparation of this list is from the lecture slides of Leow Wee Kheng [118].

- Let $\alpha$ be a scalar and $A(\alpha) \in \mathbb{R}^{n \times m}$. Derivative of $A$ with respect to $\alpha$ is defined as,

$$\frac{dA}{d\alpha} = \begin{bmatrix} \frac{dA_{11}}{d\alpha} & \frac{dA_{12}}{d\alpha} & \dots & \frac{dA_{1m}}{d\alpha} \\ \frac{dA_{21}}{d\alpha} & \dots & \dots & \frac{dA_{2m}}{d\alpha} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{dA_{n1}}{d\alpha} & \dots & \dots & \frac{dA_{nm}}{d\alpha} \end{bmatrix}$$

- Let $\alpha$ be a scalar and $A(\alpha) \in \mathbb{R}^{n \times m}$. Derivative of $A^{-1}$ with respect to $\alpha$ can be calculated as,

$$\frac{dA^{-1}}{d\alpha} = -A^{-1}\frac{dA}{d\alpha}A^{-1}$$

- Let $\alpha$ be a scalar and $A(\alpha) \in \mathbb{R}^{n \times m}$ and $B(\alpha) \in \mathbb{R}^{m \times l}$. Derivative of $AB$ with respect to $\alpha$ can be calculated as,

$$\frac{d\left(AB\right)}{d\alpha} = \frac{dA}{d\alpha}B + A\frac{dB}{d\alpha}$$

- Let $x$ be a column vector with appropriate dimension and $C \in \mathbb{R}^{n \times m}$ is independent of $x$. Then,

$$\frac{d\left(Cx\right)}{dx} = C \qquad \frac{d\left(x^T C\right)}{dx} = C^T$$

- Let $x \in \mathbb{R}^n$ be a column vector with appropriate dimension and $A \in \mathbb{R}^{n \times n}$ is independent of $x$. Then,

$$\frac{d\left(x^T A x\right)}{dx} = x^T \left(A + A^T\right)$$

- Let $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ be column vectors. Derivative of $u$ with respect to $x$ is defined as,

$$\frac{du}{dx} = \begin{bmatrix} \frac{du_1}{dx_1} & \frac{du_1}{dx_2} & \cdots & \frac{du_1}{dx_n} \\ \frac{du_2}{dx_1} & \cdots & \cdots & \frac{du_2}{dx_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{du_m}{dx_1} & \cdots & \cdots & \frac{dA_m}{dx_n} \end{bmatrix}$$

- Let $x \in \mathbb{R}^n$ is a column vector and $D \in \mathbb{R}^{m \times n}$ is a matrix with elements depend on $x$. Derivative of $Cx$ with respect to $x$ can be calculated as,

$$\frac{d\left(Dx\right)}{dx} = D + \begin{bmatrix} \frac{dD}{dx_1}x & \frac{dD}{dx_2}x & \cdots & \frac{dD}{dx_n}x \end{bmatrix}$$

*Proof.* This differentiation rule is not explicitly covered in the literature. Therefore the derivation is provided below.

Let $u = Dx$. Then in indicial notation,

$$u_i = D_{ij} x_j$$

With the definition of the derivative of a vector to another vector given in the previous bullet,

$$\frac{d\left(Cx\right)}{dx} = \left.\frac{du}{dx}\right|_{ij} = u_{i,j} = \frac{d\left(D_{ik} x_k\right)}{dx_j}$$

Also,

$$\frac{d\left(D_{ik} x_k\right)}{dx_j} = D_{ik,j} x_k + D_{ik} x_{k,j}$$

Since $x_{k,j} = \delta_{kj}$, it can be written that,

$$\frac{d\left(D_{ik} x_k\right)}{dx_j} = D_{ik,j} x_k + D_{ij}$$

First term can be formed with concatenation of the product of $x$ with derivative of the matrix $D$ with respect to every element of $x$, *i.e.*,,

$$D_{ik,j} x_k = \begin{bmatrix} \frac{dD}{dx_1}x & \frac{dD}{dx_2}x & \cdots & \frac{dD}{dx_n}x \end{bmatrix}$$

□

# APPENDIX B

# LINEARIZATION OF ROBOTIC MANIPULATOR EQUATIONS

Analysis of the boundary layer model for the robotic manipulator equations are given in Chapter 3 Section 3.3. However, derivation of the equations involve lengthy matrix manipulations and therefore they are provided in this appendix.

Collocated linearized form of the robotic manipulator equations, together with a Proportional-Derivative control law, can be written as

$$\begin{aligned}\ddot{q}_1 =& -H_{11}^{-1} \cdot \left[H_{12}\left(-K_v\dot{e} - K_p e\right) + C_{11}\dot{q}_1 + C_{12}\dot{q}_2 + G_1\right] \\ \ddot{q}_2 =& -K_v\dot{e} - K_p e\end{aligned}$$

with $e = q_2 - r_2$.

In order to clarify the mathematical manipulations, following new variable set is defined:[1]

$$x_1 = \begin{bmatrix} x_1^1 \\ x_1^2 \end{bmatrix} = \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} \qquad x_2 = \begin{bmatrix} x_2^1 \\ x_2^2 \end{bmatrix} = \begin{bmatrix} q_2 \\ \dot{q}_2 \end{bmatrix}$$

Comparing with the general form of the equations (2.6) in Chapter 2, where $\dot{x}_1 = f_1(x_1, r_2 + r + e_2)$. $x_1$ dynamics can be written as,

$$\begin{bmatrix} \dot{x}_1^1 \\ \dot{x}_1^2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -H_{11}^{-1}C_{11} \end{bmatrix} \begin{bmatrix} x_1^1 \\ x_1^2 \end{bmatrix} + \begin{bmatrix} 0 \\ -H_{11}^{-1}\left(G_1 + C_{12}x_2^2\right) \end{bmatrix} \tag{B.1}$$

For an open-chain robotic manipulator with $m$ number of passive and $n$ number of active links, $x_1 \in \mathbb{R}^{2m}$ and $x_2 \in \mathbb{R}^{2n}$.

---

[1] The usage of superscript might lead to confusion with power operator on the variables, therefore it should be reminded that no such mathematical operations are involved within this part of the derivations. The only exception is the superscript $-1$, which is used for inversion operation.

In order to analyze the boundary layer equation, linearization of the system dynamics (B.1) with respect to the $x_2$ is required. Since $x_2$ is a vector quantity,this process involves matrix differentiation. Matrix differentiation rules that are used in the derivation of the results is summarized in Appendix A. The results presented in this section are applicable to any Lagrangian system with the property,

- Inertial tensor is only function of the generalized position, and not velocity, *i.e.*,$H = H(q)$.
- Coriolis tensor is zero for zero velocity, *i.e.*,$C(q, \dot{q} = 0) = 0$

System dynamics given in (B.1), with short hand form $\dot{x}_1 = f_1$ can be separated into two as $\dot{x}_1^1 = f_1^1$ and $\dot{x}_1^2 = f_1^2$. Then the matrix derivative $\frac{\partial f_1}{\partial x_2}$ can be written as,

$$\frac{\partial f_1}{\partial x_2} = \begin{bmatrix} \frac{\partial f_1^1}{\partial x_2^1} & \frac{\partial f_1^1}{\partial x_2^2} \\ \frac{\partial f_1^2}{\partial x_2^1} & \frac{\partial f_1^2}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \frac{\partial f_1^2}{\partial x_2^1} & \frac{\partial f_1^2}{\partial x_2^2} \end{bmatrix}$$

With this form of equations, $f_1^2$ can be written as,

$$f_1^2 = -H_{11}^{-1} C_{11} x_1^2 - H_{11}^{-1} G_1 - H_{11}^{-1} C_{12} x_2^2 \tag{B.2}$$

Each of the three terms of (B.2) are column vectors with dimension $m$. Derivatives of these terms to $n$ dimensional column vectors $x_2^1$ and $x_2^2$ are derived below.

- $\left(H_{11}^{-1} C_{11} x_1^2\right)$ :

$$\begin{aligned} \frac{\partial\left(H_{11}^{-1} C_{11} x_1^2\right)}{\partial x_2^1} &= \frac{\partial H_{11}^{-1}}{\partial x_2^1} C_{11} x_1^2 + H_{11}^{-1} \frac{\partial C_{11}}{\partial x_2^1} x_1^2 \\ &= \left(-H_{11}^{-1} \frac{\partial H_{11}}{\partial x_2^1} H_{11}^{-1} C_{11} + H_{11}^{-1} \frac{\partial C_{11}}{\partial x_2^1}\right) x_1^2 \\ \frac{\partial\left(H_{11}^{-1} C_{11} x_1^2\right)}{\partial x_2^2} &= \underbrace{\frac{\partial H_{11}^{-1}}{\partial x_2^2}}_{=0, \text{ since } H \text{ is independent of } \dot{q}} C_{11} x_1^2 + H_{11}^{-1} \frac{\partial C_{11}}{\partial x_2^2} x_1^2 = H_{11}^{-1} \frac{\partial C_{11}}{\partial x_2^2} x_1^2 \end{aligned}$$

- $\left(H_{11}^{-1} G_1\right)$:

$$\begin{aligned} \frac{\partial\left(H_{11}^{-1} G_1\right)}{\partial x_2^1} &= \frac{\partial H_{11}^{-1}}{\partial x_2^1} G_1 + H_{11}^{-1} \frac{\partial G_1}{\partial x_2^1} = -H_{11}^{-1} \frac{\partial H_{11}}{\partial x_2^1} H_{11}^{-1} G_1 + H_{11}^{-1} \frac{\partial G_1}{\partial x_2^1} \\ \frac{\partial\left(H_{11}^{-1} G_1\right)}{\partial x_2^2} &= H_{11}^{-1} \frac{\partial G_1}{\partial x_2^2} \end{aligned}$$

- $\left(H_{11}^{-1}C_{12}x_2^2\right)$ :

$$\begin{aligned}\frac{\partial\left(H_{11}^{-1}C_{12}x_2^2\right)}{\partial x_2^1} &= \left[\frac{\partial H_{11}^{-1}}{\partial x_2^1}C_{12} + H_{11}^{-1}\frac{\partial C_{12}}{\partial x_2^1}\right]x_2^2 \\ &= -H_{11}^{-1}\frac{\partial H_{11}}{\partial x_2^1}H_{11}^{-1}C_{12}x_2^2 + H_{11}^{-1}\frac{\partial C_{12}}{\partial x_2^1}x_2^2 \\ \frac{\partial\left(H_{11}^{-1}C_{12}x_2^2\right)}{\partial x_2^2} &= H_{11}^{-1}\frac{\partial\left(C_{12}x_2^2\right)}{\partial x_2^2} = H_{11}^{-1}\left(C_{12} + \frac{\partial C_{12}}{\partial x_2^2}x_2^2\right)\end{aligned}$$

With these expressions, matrix derivative of $f_1$ can be evaluated with,

$$\begin{aligned}\frac{\partial f_1^2}{\partial x_2^1} = &-\left(-H_{11}^{-1}\frac{\partial H_{11}}{\partial x_2^1}H_{11}^{-1}C_{11} + H_{11}^{-1}\frac{\partial C_{11}}{\partial x_2^1}\right)x_1^2 \\ &-\left(-H_{11}^{-1}\frac{\partial H_{11}}{\partial x_2^1}H_{11}^{-1}G_1 + H_{11}^{-1}\frac{\partial G_1}{\partial x_2^1}\right) \\ &-\left(-H_{11}^{-1}\frac{\partial H_{11}}{\partial x_2^1}H_{11}^{-1}C_{12} + H_{11}^{-1}\frac{\partial C_{12}}{\partial x_2^1}\right)x_2^2\end{aligned} \tag{B.3}$$

and

$$\begin{aligned}\frac{\partial f_1^2}{\partial x_2^2} = &-H_{11}^{-1}\frac{\partial C_{11}}{\partial x_2^2}x_1^2 \\ &- H_{11}^{-1}\frac{\partial G_1}{\partial x_2^2} \\ &- H_{11}^{-1}\left(C_{12} + \frac{\partial C_{12}}{\partial x_2^2}x_2^2\right)\end{aligned} \tag{B.4}$$

**Linearization around** $\dot{q}_2 = 0$

Fault tolerant control of robotic manipulators involve linearization around the point $(q_2, \dot{q}_2) = (r_2, 0)$. Since $C(q, \dot{q} = 0) = 0$, Jacobians given in (B.3) and (B.4) can be simplified $C = 0$. Furthermore, if damping in the joints are neglected, $G_1$ becomes independent of joint velocities, *i.e.*,$G_1 = G_1(q)$ and hence it can be written that $\frac{\partial G_1}{\partial x_2^2} = 0$.

With these assumptions, equations (B.3) and (B.4) can be written as,

$$
\begin{aligned}
\left.\frac{\partial f_1^2}{\partial x_2^1}\right|_{x_2=(r_2,0)} &= \left.\left(H_{11}^{-1}\frac{\partial H_{11}}{\partial x_2^1}H_{11}^{-1}G_1 - H_{11}^{-1}\frac{\partial G_1}{\partial x_2^1}\right)\right|_{q_2=r_2} \\
\left.\frac{\partial f_1^2}{\partial x_2^2}\right|_{x_2=(r_2,0)} &= 0
\end{aligned}
$$

and therefore

$$
\left.\frac{\partial f_1}{\partial x_2}\right|_{x_2=(r_2,0)} = \begin{bmatrix} 0 & 0 \\ \left.\left(H_{11}^{-1}\frac{\partial H_{11}}{\partial x_2^1}H_{11}^{-1}G_1 - H_{11}^{-1}\frac{\partial G_1}{\partial x_2^1}\right)\right|_{q_2=r_2} & 0 \end{bmatrix} \tag{B.5}
$$

# CURRICULUM VITAE

## PERSONAL INFORMATION

**Surname, Name:** Marangoz, Alp
**Nationality:** Turkish (TC)
**Date and Place of Birth:** 27.06.1980, Ankara
**Marital Status:** Married
**e-mail:** alpmarangoz@yahoo.com

## EDUCATION

| Degree | Institution | Year of Graduation |
|---|---|---|
| Master of Science | METU Aerospace Engineering | 2005 |
| Bachelor of Science | METU Aeronautical Engineering | 2002 |
| High School | Ankara Atatürk Lisesi | 1998 |

## PROFESSIONAL EXPERIENCE

| Year | Place | Enrollment |
|---|---|---|
| 2007- | Roketsan A. Ş. | Senior Engineer |
| 2005-2006 | Von Karman Inst. for Fluid Dynamics | Diploma Course Member |
| 2002-2005 | Roketsan A. Ş. | Systems Engineer |

## PUBLICATIONS

1. Alp Marangoz and Ali Türker Kutay. “Eksik eyleyicili kararsız iki serbestlik dereceli robot kolun uyarlamalı tersini alma yöntemi ile kontrolü”. In: Otomatik

Kontrol Ulusal Toplantısı (TOK’2017), (2017). p. 392-397.

2. Alp Marangoz and Ali Türker Kutay. “A control system architecture for control of non-affine in control, open-loop unstable underactuated systems”. In: Mechanical and Aerospace Engineering (ICMAE), 2017 8th International Conference on. IEEE, (2017), pp. 780-785.
3. Alp Marangoz. “Süpersonik akıma dik sıvı enjekte edilmesinin deneysel olarak incelenmesi”. In: Savunma Teknolojileri Kongresi (SAVTEK’2008), (2008). p. 29.