Development of Fault Diagnosis and Fault Tolerant Control Algorithms with Application to Unmanned Systems

M. Hadi Amoozgar

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

August 2012

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis proposal prepared

By: **M. Hadi Amoozgar**

Entitled: **Development of Fault Diagnosis and Fault Tolerant Control Algorithms with Application to Unmanned Systems**

and submitted in partial fulfilment of the requirements for the degree of

Master of Applied Science(Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

________________________________ Chair
Dr. O. Kuzgunkaya

________________________________ Examiner
Dr. W. F. Xie

________________________________ Examiner
Dr. A. G. Aghdam ECE External

________________________________ Supervisor
Dr. Y. M. Zhang

Approved by: ________________________________
Dr. S. Narayanswamy, MASc Program Director
Department of Mechanical and Industrial Engineering

____
Dean Robin Drew
Faculty of Engineering & Computer Science

Date: _________

ABSTRACT

Development of Fault Diagnosis and Fault Tolerant Control Algorithms with Application to Unmanned Systems

Hadi Amoozgar, M.A.Sc.
Concordia University, 2012

Unmanned vehicles have been increasingly employed in real life. They include unmanned air vehicles (UAVs), unmanned ground vehicles (UGVs), unmanned spacecrafts, and unmanned underwater vehicles (UUVs). Unmanned vehicles like any other autonomous systems need controllers to stabilize and control them. On the other hand unmanned systems might subject to different faults. Detecting a fault, finding the location and severity of it, are crucial for unmanned vehicles. Having enough information about a fault, it is needed to redesign controller based on post fault characteristics of the system. The obtained controlled system in this case can tolerate the fault and may have a better performance. The main focus of this thesis is to develop Fault Detection and Diagnosis (FDD) algorithms, and Fault Tolerant Controllers (FTC) to increase performance, safety and reliability of various missions using unmanned systems.

In the field of unmanned ground vehicles, a new kinematical control method has been proposed for the trajectory tracking of nonholonomic Wheeled Mobile Robots (MWRs). It has been experimentally tested on an UGV, called *Qbot*. A stable leader-follower formation controller for time-varying formation configuration of multiple nonholonomic wheeled mobile robots has also been presented and is examined through computer simulation.

In the field of unmanned aerial vehicles, Two-Stage Kalman Filter (TSKF), Adaptive Two-Stage Kalman Filter (ATSKF), and Interacting Multiple Model (IMM) filter were proposed for FDD of the quadrotor helicopter testbed in the presence of actuator faults. As for space missions, an FDD algorithm for the attitude control system of the Japan Canada Joint Collaboration Satellite - Formation Flying (JC2Sat-FF) mission has been developed. The FDD scheme was achieved using an IMM-based FDD algorithm. The efficiency of the FDD algorithm has been shown through simulation results in a nonlinear simulator of the JC2Sat-FF.

A fault tolerant fuzzy gain-scheduled PID controller has also been designed for a quadrotor unmanned helicopter in the presence of actuator faults. The developed FDD algorithms and fuzzy controller were evaluated through experimental application to a quadrotor helicopter testbed called *Qball-X4*.

Dedicated to my lovely parents,

Ashraf and **Hossein**

for their pure love, endless support
and encouragement.

ACKNOWLEDGEMENTS

It would not have been possible to write this thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

Above all, I would like to thank my parents for their endless support and great patience. My brother, Ali, and my sister, Marzieh, who have supported me throughout, as always, for which my mere expression of thanks likewise does not suffice.

I am grateful to my supervisor, Professor Youmin Zhang, for his support and guidance throughout my work. This work would have never been done without his advice and comments. I also highly appreciate him for being infinitely genuine, kind and considerate towards me.

With great pleasure, I express my gratitude to my colleagues, Dr. Abbas Chamseddine and Dr. Narendra Gollu, whom I had the privilege to closely work with. Without their guidance I could not finish this study.

Last, but not least, my special thanks to my friends, Iman, Mahyar, Ensieh, Hossein and Alireza Mehrabian, whom their help and encouragement motivated me during the course of this thesis work. I am also grateful to all those people who have given me support and I didn't have the chance to thank them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Motivation

Unmanned vehicles have been increasingly employed in real life. They become more and more popular in military and civil applications. These autonomous unmanned vehicles include unmanned air vehicles (UAVs), unmanned ground vehicles (UGVs), unmanned spacecrafts, and unmanned underwater vehicles (UUVs).

Reliability, maintainability and survivability of unmanned systems have been the issue of many researches over the last decades. Unmanned vehicles like any other autonomous systems need controllers to stabilize and control them. On the other hand unmanned systems might subject to different faults. Detecting a fault, finding the location and severity of it, are crucial for unmanned vehicles. Having enough information about a fault, it is needed to redesign controller based on post fault characteristics of the system. The obtained controlled system in this case can tolerate the fault and may have a better performance.

In the area of Unmanned Ground Vehicles (UGV), different methods have been proposed for the trajectory tracking of wheeled mobile robots. In kinematical control level, determination of the most appropriate heading angle of the robot is one of the fundamental problems in the trajectory tracking of WMRs. In some researches such as [2], a pursuit guidance law is used to determine appropriate heading angle of the vehicle during operation. Although pursuit guidance is straightforward, the tracking performance is not always satisfactory. In [3] a new fuzzy scheduler has been devised to compute appropriate heading angle of the vehicle instantly, in addition to two fuzzy controllers which separately control linear and angular velocities. The main drawback of the aforementioned research work is the lack of explicit mathematical method to prove stability of the proposed a set of fuzzy controllers. Motivated by this fact, as a part of work in this thesis, a new stable controller is developed for trajectory tracking of mobile robots.

Various studies have also been done in both theoretical and application aspects for the robots formation. Different approaches and strategies have been proposed for the formation control of multiple robots. Typically, using a more strict coordination strategy attempts to provide the desired configuration of the robotic group. The ability to change configuration is crucial to group of robots in formation, for example before the group passes through a corridor, the robots should be aligned. In lots of researches, switching between two configurations is discussed, for example wedge to column, or line to wedge. Very few numbers of researches focused on the transition period between two configurations, while the performance of the group may drastically decrease during the transition. This was the motivation to design a stable leader-follower formation controller for time-varying formation configuration of nonholonomic wheeled mobile robots.

In the area of Unmanned Aerial Vehicles (UAV) and over the last two decades, reliability, maintainability and survivability of UAVs have drawn significant attention into Fault Tolerant Control (FTC) and Fault Detection and Diagnosis (FDD) problems. Accurate information about the time, location and severity of the fault help designers to reconfigure the control structure and will help them to avoid system's unexpected shut down, break down or even facility damages in the event of the fault. One of the key challenges in this area is to design an FDD scheme which is highly sensitive to faults and less sensitive to external disturbances. In some research works on FTC, it has been assumed that perfect information of the fault is available [4] while such an assumption may not be realistic in real world applications. Indeed, to design a reconfigurable fault-tolerant controller, the FDD scheme should provide detailed information of the post-fault system as accurate as possible [5]. This fact became a motivation to implement and verify some FDD algorithms with application to rotary-wing UAV available at the Diagnosis, Flight Control and Simulation Lab (DFCSL) of the Department of Mechanical and Industrial Engineering at Concordia University in this thesis.

On the other hand, efficient FDD of faulty components in satellite missions can also significantly increase the spacecraft reliability and has been the subject of interests in recent years [6, 7]. This is more appreciated in case of satellite formation flying missions because of the more stringent safety requirements when a group of spacecrafts fly in close proximity. A part of the current thesis is dedicated to move a step forward towards the effective fault protection of the satellites. Specifically, it will present a FDD algorithm for the momentum wheels (MWs) of the attitude control system of the satellites.

One of the other objectives of this thesis is to propose fault tolerant control methods that are effective, simple to be implemented for real-time applications and robust to model uncertainties and external disturbances including actuator faults. PID (Proportional - Integral - Derivative) controllers are the most well-known controller in the society of automation and control, due to their simple structure and wide variety of usages. These kinds of controllers are classified into two main categories in terms of parameters selection strategies. In the first group, controller gains are fixed during operation while in the second group, gains change based on the operating conditions. In the first group, gains are tuned by the designer and remain invariable during the operation. One of the most well-known methods for choosing control gains in this group is Ziegler-Nichols method which has been addressed in lots of research works [8]. Although this method is simple and straightforward, fine tuning is required for different applications. In most applications, due to structural changes the controlled system may lose its effectiveness, therefore the PID gains need to be continuously retuned during the system life span. To reduce the effort of retuning the gains and also in order to increase system's performance, an adaptive gain scheduler based on fuzzy inference system is developed for application to rotary-wing UAV in this thesis.

1.2 Literature Review

This section presents a review of the relevant literature on Unmanned Systems. First a review on trajectory tracking control of wheeled mobile robots will come, followed by a brief review on formation control of multiple mobile robots. Then a review of different Fault Tolerant Control (FTC) and Fault Detection and Diagnosis (FDD) algorithms with application to aircrafts and spacecrafts will be presented.

Mobile robots, like unicycle and car-like vehicles, have considerably wide range of applications including discovery, observation, search and rescue, and mapping of unknown or partially known environments [9] (Fig. 1.1). One of the difficulties in the control of mobile robots lies in the fact that ordinary vehicles possess only two degrees of freedom (linear velocity and rotational velocity) for locomotion control, whereas vehicles have three degrees of freedom, x, y and θ in its positioning [10] (Fig. 1.2). These kinds of mobile robots are classified as underactuated systems.

Under the assumption of no slippage in robot's base, the system will be imposed upon by a nonholonomic constraint. Nonholonomic control problems are quite challenging making them an attractive research area in control fields [11]. A complete study of nonholonomic control problems have been presented in [12], [13].

During the past 30 years extensive research works have been devoted to the problem of trajectory tracking control of nonholonomic WMRs. In majority of these researches the control inputs are obtained by a combination of feedforward inputs, calculated from a

Figure 1.1: Different applications of wheeled mobile robots.

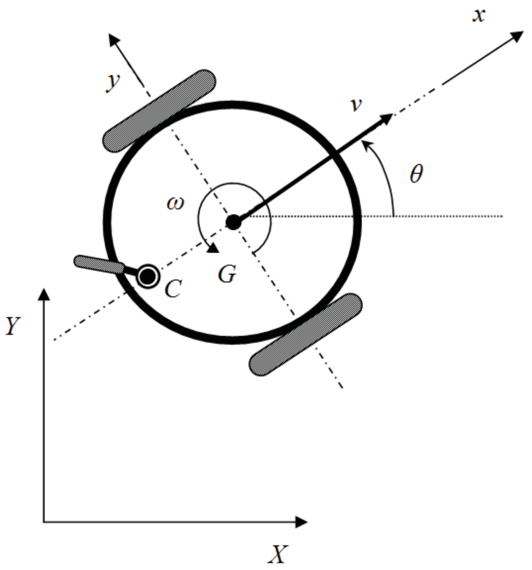


Figure 1.2: The schematic of the mobile robot.

reference trajectory, and a feedback control law, as in [14].

Lyapunov stable, time-varying, state-tracking control laws were presented in [10]. Many variations and improvements of this simple and effective state-tracking controller were followed by latter research works [14]. An adaptive extension of this work was introduced in [15], where adaptive capabilities are included to increase the robustness to robot modelling uncertainties.

A model predictive controller based on a linearized error dynamics was presented in [16], which minimizes the difference between the future trajectory-following errors of the robot and the reference robot. Reference [17] presents a nonlinear model predictive controller to solve the problem of combined trajectory tracking and path following. An adaptive controller in the presence of unknown skidding and slipping is designed in [18], where nonholonomic constraints assumed to be perturbed. In [19] an adaptive controller has been developed in the presence of uncertainties in both dynamic and kinematic parameters of a mobile robot.

In addition, various control methods such as backstepping [20], reinforcement learning [21], transfer function approach [22], neural network [23, 24], sliding mode [25], fuzzy logic [3] have also been exploited for the control of WMRs.

In later research studies combinations of these methods are also considered. Dual adaptive neural network controller is designed in [26]. Genetic algorithm was used to optimize parameters of fuzzy controllers, as in [27, 28].

Formation control problem simply means the problem of controlling the relative positions and orientations of robots in a group while allowing the group to move as a whole [29, 30]. There are many potential advantages of a group of robots over a single robot, including greater flexibility, adaptability, robustness, sharing the sensor data, and robot parallelism [31]. In addition, added interests will result in the cooperating teams when the tasks may be inherently too complex for a single system to accomplish. This can help build and use several simpler and more flexible, fault-tolerant or cheaper systems rather than using one single and large system [32]. Numerous application issues, such as search and rescue [33, 34], intelligent automatic navigation systems in highways [35], air traffic control [36], and soccer robots can be found, where the use of several robots as a coordinated team is more effective than those which work separately without any coordination. In addition, there are various applications of formation control of a collection of robots such as military applications, field surveillance and exploration [37], application in hazard situations, mapping of unknown or partially known environments, distributed manipulation, and transportation of large objects [29].

These benefits have put forward an active field for some researchers of robotics community in recent years to examine the solution of this challenging problem. Various studies have been done in both theoretical and applied aspects for the robots formation. Various approaches and strategies have been proposed for the formation control of multiple robots. Typically, using a more strict coordination strategy attempts to provide the desired configuration of the robotic group. These approaches can be roughly categorized into three main methods: virtual structure, behavior-based approach, and leader-follower, each of which has advantages and weaknesses.

Some research works use virtual structure approach in the formation control [38, 39, 40, 41]. The virtual structure scheme considers the whole formation as a single virtual rigid arrangement. Desired motion is assigned to the virtual structure as a group, which in turn results into the trajectories for each robot in the formation to follow. The main advantages of the virtual structure approach is that it is fairly easy to prescribe the coordinated behavior for the group, and the formation can be maintained very well during the maneuvers, that is, the virtual structure can evolve as a whole in a given direction with some given orientation and maintain a rigid geometric relationship among multiple vehicles [42]. The main disadvantage of the virtual structure implementation is the centralization, which leads a single point of failure for the whole system [43]. Furthermore, if the formation has to maintain the exact same virtual structure all the time, the potential applications are limited, especially when the formation shape is time-varying or needs to be frequently reconfigured [42].

In the behavior-based scheme, a number of desired activities are considered for each robot, and then the final behavior of each robot is determined by weighting the comparative importance of each behavior. Several probable behaviors contain obstacle avoidance, collision avoidance, goal searching and formation maintenance [44, 45]. The advantage is that it is natural to derive control strategies when vehicles have multiple competing objectives, and an explicit feedback is included through communication between neighbors [42]. The main drawback of the behavior-based approach is that due to the lack of explicit dynamical functions involved, the formation performance cannot be analyzed mathematically [46].

In the leader and follower method [47, 48], one of the robot members of the group is considered to be the leader and the other members are expected to follow the leader. Consequently, the formation control problem converts into two simple problems including trajectory tracking by the leader robot and control/maintenance of the formation by the rest of the components. There may be cases where a robot can be a leader of another robot while it may be the follower of a different one. In order to keep the desired formation during a maneuver, the follower robots should adjust their positions relative to the leader robot. Proper controllers can be designed in order that the desired relative positions between the leader and the followers are generated.

In the leader-follower method, the position of the follower is determined with respect to the leader robot by either a distance-angle ($l-\varphi$) or a distance-distance ($l-l$) model (see Fig. 1.3). In the distance-angle model, the robots are controlled like the rings of a chain in such a way that each robot simply follows another single robot. In the distance-distance model, each robot follows two other robots simultaneously. The arrangement of the robots can be made in various shapes such as a line, column, diamond, and a wedge.

It should be pointed out that the simplicity of mathematical analysis and a higher safety measures for the motion of the collection of the robots as well as keeping the formation configuration are the main advantages of the leader-follower approach. In addition, the formation can still be maintained even if the leader is perturbed by some disturbances [42].

Due to the above mentioned advantages of leader-follower approach, a number of researches have focused on this method [47, 48, 49, 50, 51, 52]. The main disadvantage of the leader-follower method is the necessity of availability of leader's controlled input to the

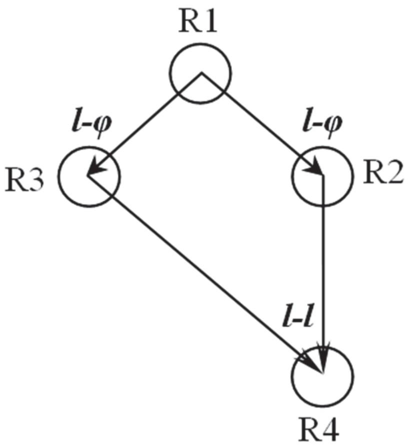


Figure 1.3: Distance-angle and distance-distance models.

follower for each pair of leader and follower [46]. Also, there is no explicit feedback to the formation, that is, there is no explicit feedback from the followers to the leader in this case [42].

Reference [49] designed the backstepping leader-follower formation controller to release the perfect velocity assumption at the dynamics level. In [50] an adaptive leader-follower formation controller was presented which does not require velocity information of the leader robot. It used smooth projection algorithm to estimate the unknown velocities of the leader robot. The leader's trajectory constraints in the presence of constraints for followers have been discussed in [51], so that the followers can follow the relative leaders while respecting their input constraints. In [52], two model predictive controllers were proposed for both formation keeping and obstacle avoidance.

Relatively small number of researches addressed the problem of fault detection of helicopter [53, 54]. Heredia *et al.* [55, 56] used a simple observer to detect and diagnose the fault in helicopter's actuator. Differential flatness techniques are employed to detect and diagnose sensor faults in [57] and actuator faults in [58]. The approach introduces non-linear observers and takes profit of differential flatness of the system dynamics to design a fault detection analytical redundancy scheme. Observer-based FDD techniques are the most common in the literature: in [59], Thau's observer is employed to generate a set of residuals for detection and diagnosis of accelerometers and inclinometers faults. Thau's observer is also employed in [60] for actuator faults. In [61], a set of N sensors (accelerometers, magnetometers and rate gyros) is embedded in the quadrotor. For the N available measurements, $N+1$ non-linear observers are designed. The first observer uses the whole set of measurements (i.e. N measurements), while the remaining observers use a subset of sensors. Therefore, these N observers are insensitive to faults in the discarded sensors. A fault-tolerant switching is then designed for the quadrotor system under consideration. At each instant of time, the attitude estimate that presents the smallest error when compared to the reference attitude is selected to be fed to the control law. In [62], the problem of simultaneous observer based sensor diagnosis and speed estimation of the quadrotor UAV is investigated. The main features lie in the use of a useful bank of reduced order observers to detect and isolate faulty sensors and at the same time to provide unbiased speed estimation of UAV from accelerometers. The work in [63] compares two diagnostic techniques applied to MEMS sensors of an Inertial Measurement Unit (IMU). The first approach is based on parameter estimation with nonlinear optimization technique while the second one makes use of set membership estimation. Both techniques are applied to the detection of faults in the IMU fixed on a quadrotor in quasi-static movement.

Number of fault protection techniques has been presented in literature for spacecraft. Morgan [6] discussed detailed protection techniques that are general to different types of spacecraft. Olive [7] discussed fault protection techniques for next generation of autonomous satellites which include such research tracks as active diagnosis and hybrid diagnosis. Ruiter et. al [64] developed a fault tolerant controller for magnetic torque rods in the presence of partial failure of the magnetorquers. In regard to fault protection of momentum wheels, Tehrani et.al [65] used model-based fault detection methods and artificial intelligence techniques to detect and isolate faults. Reference [66] uses artificial intelligence techniques to detect and isolate faults. For a complete survey on the methods of fault protection techniques for momentum wheels the readers are referred to [67].

Two-Stage Kalman Filter (TSKF) proposed by Keller and Darouach in [68]. In [69], the authors proposed the use of forgetting factor technique (which is widely used in system identification applications) for TSKF. As an extension, in [70, 71] an adaptive algorithm is added to the TSKF to make it more responsive to abrupt changes in the control effectiveness factors. The Interacting Multiple Model (IMM) method is one of the most efficient approaches for FDD applications, which was first published in [72]. By extending the Interacting Multiple Model (IMM) based FDD scheme to nonlinear systems, Unscented Kalman Filter (UKF) is combined with the IMM for detecting sensor faults in fixed-wing UAVs in [73]. A Dual Unscented Kalman Filter (DUKF) was also applied to a NASA fixed-wing Generic Transport Model (GTM) UAV in [74].

Gain-scheduled controller can also be categorized as Fault Tolerant Control (FTC) like Gain-scheduled PID Controllers. Several methods have been proposed in the literature for gain scheduling PID. In [75] a stable gain-scheduling PID controller is developed based on grid point concept for nonlinear systems, in which gains switch between some predefined values. Different gain scheduling methods were studied and compared in [76]. In [77] a new PID scheme is proposed in which the controller gains were scheduled by a fuzzy inference scheme. Many variations and improvements of this simple and effective method were followed by latter research works [78, 79, 80]. A particle swarm optimization method is used in [78] to design membership functions of fuzzy PID controller. In [81], an accumulated genetic algorithm is proposed which learns the parameters and number of fuzzy rules in the fuzzy PID controller. An adaptive fuzzy PID using neural wavelet network is presented in [82]. The interested readers can find a brief review of different fuzzy PID structures in [83].

A Gain-Scheduled PID (GS-PID) is designed for the quadrotor system in [84]. The GS-PID has been implemented for different sections of the entire flight envelope by properly tuning the PID controller gains for both normal and fault conditions. The switching from one PID to another is then based on the actuator's health status. It is worthy to note that the above method requires a Fault Detection and Diagnosis (FDD) scheme to provide the time of fault occurrence as well as the location and the magnitude of the fault during the flight.

1.3 Problem Formulation

Based on the state of the art and advantages and disadvantages of the existing methods discussed in the previous section, this section formulates the main objectives of the thesis and original contributions.

1.3.1 Thesis Objectives

The main objective of the presented thesis is to develop fault detection and diagnosis algorithms and fault tolerant controllers to enhance performance, safety and reliability of various missions using unmanned systems.

As one of the objectives of the current thesis, new control methodology has been developed for trajectory tracking of nonholonomic wheeled mobile robot which makes the robot catch its desired trajectory faster with less control efforts.

Another objective is to design a stable controller for group of multiple vehicles in time-varying formation configurations.

To increase reliability of quadrotors, different Fault Detection and Diagnosis and Fault Tolerant Control approaches have been developed for the presence of actuator faults.

Attitude Control System (ACS) is one of the main satellite subsystems, which is to detect, estimate, and control the orientation of the satellite. One of the objectives is to present FDD algorithms for ACS actuators, which are the components used to generate torques. The types of ACS actuators frequently seen in practice include reaction/momentum wheels, magnetorquers, and thruster pairs. In this work, the FDD algorithms are developed for momentum wheels. It is our intention that the FDD algorithms shall be integrated into spacecraft on-board software and be able to autonomously detect and identify hardware failures, and in addition, to autonomously *reconfigure* the attitude control system if necessary. To get our research into perspective of practical missions, the scenarios of a spacecraft formation flying mission, called JC2Sat-FF [85], will be used which is introduced in more details later on.

1.3.2 Case Studies

For the purpose of testing and validation of the proposed methods and algorithms, different testbeds have been used.

Quadrotor Helicopter: The quadrotor helicopter is relatively a simple, affordable and easy to fly system. It has been widely used to develop, implement and flying-test methods in control, fault diagnosis, fault tolerant control as well as multi-agent based technologies in formation flight, cooperative control, distributed control, surveillance and search missions, mobile wireless networks and communications. Some theoretical works consider the problems of control [86], formation flight [87] and fault diagnosis [62] of the quadrotor Unmanned Aerial Vehicle (UAV). However, few research laboratories are carrying out advanced theoretical and experimental works on the system. Among others, one may cite for example, the UAV health management project of the Aerospace Controls Lab. at MIT [88], the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control project [89] and the Micro Autonomous Systems Technologies project [90]. A team of researchers is also currently working at the Department of Mechanical and Industrial Engineering of Concordia University to develop, implement and test approaches in Fault Detection and Diagnosis (FDD), Fault Tolerant Control (FTC) and cooperative control with experimental application to the quadrotor unmanned helicopter system. For more information on the research activities carried out, interested readers are referred to the Networked Autonomous Vehicles (NAV) laboratory [91].

Wheeled Mobile Robot: As a testbed for wheeled mobile robot's related applications a robot vehicles called “Qbot” is used. Qbot ia a multi-purpose research robotic developed by Quanser Inc. Qbot has two differentially driven wheels, and a caster wheel (to maintain its stability).

Satellite Mission: As a testbed for spacecraft applications JC2Sat is chosen. The JC2Sat-FF is a joint Canadian Space Agency/Japan Aerospace Exploration Agency technology mission consisting of two nano-satellites and weighing approximately at 18kg each. The main objective of this mission is to demonstrate the feasibility of maintaining along-track spacecraft formation by using only differential atmospheric drag control between the two satellites and GPS-based relative navigation (Fig. 2.6).

1.3.3 Original Contributions

This part summarizes the contributions of the presented thesis as follows.

- Inspired by logics which have been used in [3], a new control structure for trajectory tracking of mobile robots, named Lyapunov-based Guidance, is presented in this thesis. The performance of the proposed control method is compared to that of Model Predictive Control (MPC), Linear State Tracking Control (LSTC) and Nonlinear State Tracking Control (NSTC) methods in terms of tracking performance and control effort, through real time application to *Qbot* testbed. It is also proved that the controlled system with the proposed controller is stable.

- A stable leader-follower formation controller for time-varying formation configuration of nonholonomic wheeled mobile robots is presented. Separation-bearing is used to describe relative coordination of leader-follower pairs of robots. It’s assumed that the transition between two configurations is described as a function of time in both separation distance and bearing angle. A virtual follower is used to specify the desired coordination of the real follower, and then a controller is designed to make the follower robot as close as possible to its virtual one.

- A Two-Stage Kalman Filter (TSKF), an Adaptive Two-Stage Kalman Filter (ATSKF) and an Interacting Multiple Model (IMM) have been implemented on the quadrotor helicopter for detection, isolation and identification of actuator faults.

- As further extension, IMM-based FDD [72] is also applied to the nonlinear dynamics of the satellite. The IMM-based FDD scheme has the advantage of not only detecting faults but also providing the information on location and magnitude of the fault. For partial faults, the magnitude can be determined by the probabilistically weighted sum of the fault magnitudes of the corresponding partial fault model. In addition, FDD is integrated with state estimation. As mentioned above, these proposed FDD schemes are illustrated via a nano-satellite formation flying mission, JC2Sat-FF. However, it should be emphasized that the proposed methodologies are rather generic and can be readily extended and adapted to the cases of other types of momentum wheels found in space missions.

- An adaptive PID controller is also proposed for fault tolerant control of a quadrotor helicopter system. A fuzzy inference scheme is used to tune in real-time the controller gains, where the tracking error and the change in tracking error are used in this fuzzy scheduler to make the system act faster and more effectively in the fault-free case as well as in the event of fault occurrence. The proposed PID controller is compared with the conventional one through an experimental application to the quadrotor helicopter testbed at the NAV Lab.

1.3.4 Thesis Outline

The outline of the thesis is given below.

- **Chapter 1:** The introduction, contributions and thesis outline are presented in this chapter.

- **Chapter 2:** Description and dynamic equations of the testbeds are presented in this chapter.

- **Chapter 3:** This chapter presents three Fault Detection and Diagnosis algorithms for the purpose of actuator fault detection and diagnosis in unmanned systems.

- **Chapter 4:** This chapter presents some control algorithms including fault tolerant control for unmanned systems.

- **Chapter 5:** In this chapter simulation and experimental results of developed methods and algorithms are presented.

- **Chapter 6:** This chapter concludes the thesis and makes some recommendations for future work.

1.4 Publications

The following publications were written during the course of the thesis work.

1. **M. H. Amoozgar**, A. Chamseddine, and Y. M. Zhang, “Experimental Test of a Two-Stage Kalman Filter for Actuator Fault Detection and Diagnosis of an Unmanned Quadrotor Helicopter,” Accepted for publishing in The Journal of Intelligent and Robotic Systems (JINT), Springer, July, 2012 [ID: JINT-D-12-00148].

2. A. Chamseddine, **M. H. Amoozgar**, and Y. M. Zhang, “Experimental Validation of Fault Detection and Diagnosis for Unmanned Aerial Vehicles,” Book Chapter Submitted to Handbook of Unmanned Aerial Vehicles, Springer, May, 2012.

3. **M. H. Amoozgar**, A. Chamseddine, and Y. M. Zhang, “Experimental Test of an Interacting Multiple Model Filtering Algorithm for Actuator Fault Detection and Diagnosis of an Unmanned Quadrotor Helicopter,” Accepted by the International Conference on Intelligent Robotics and Applications (ICIRA12), October, 2012, Montreal,

Quebec, Canada.

4. **M. H. Amoozgar**, A. Chamseddine, and Y. M. Zhang, “Fuzzy Gain-Scheduled PID for Payload Drop: Application to an Unmanned Quadrotor Helicopter Testbed,” Accepted by the International Conference on Intelligent Unmanned Systems (ICIUS 2012), October, 2012, Singapore.

5. **M. H. Amoozgar** and Y. M. Zhang, “Varying-Configuration Formation Control of Multiple Wheeled Mobile Robots,” Accepted by the International Conference on Intelligent Unmanned Systems (ICIUS 2012), October, 2012, Singapore.

6. **M. H. Amoozgar**, Y. M. Zhang, J. Lee, and A. Ng, “A Fault Detection and Diagnosis Technique for Spacecraft in Formation Flying,” Accepted by the IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (Safeprocess12), August, 2012, Mexico City, Mexico.

7. **M. H. Amoozgar** and Y. M. Zhang, “Trajectory Tracking of Wheeled Mobile Robots: A Kinematical Approach,” IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA12), July, 2012, Suzhou, China.

8. **M. H. Amoozgar**, A. Chamseddine and Y. M. Zhang, “Experimental Test of a Two-Stage Kalman Filter for Actuator Fault Detection and Diagnosis of an Unmanned Quadrotor Helicopter,” Proceedings of the IEEE 2012 International Conference on Unmanned Aircraft Systems, June, 2012, Philadelphia, USA.

9. **M. H. Amoozgar**, A. Chamseddine, Y. M. Zhang, “Fault-Tolerant Fuzzy Gain-Scheduled PID for a Quadrotor Helicopter Testbed in the Presence of Actuator Faults,” Proceedings of the IFAC Conference on Advances in PID Control, March, 2012, Brescia, Italy.

10. **M. H. Amoozgar**, N. Gollu, Y. M. Zhang, J. Lee, and A. Ng, “Fault Detection and Diagnosis of Attitude Control System for the JC2Sat-FF Mission,” Proceedings of the

4th International Conference on Spacecraft Formation Flying Missions and Technology, May, 2011, Saint Hubert, QC, Canada.

1.5 Chapter Summary

In this chapter, the thesis objectives and the state of the arts were discussed. The next chapter will discuss the description and dynamics the testbeds models.

Chapter 2

Testbeds Description and Dynamics

This chapter is dedicated to describe the testbeds models used in the current research. First a brief description of a quadrotor helicopter called *Qball-X4* comes, followed by its subsystems, configuration and mathematical dynamic model. Then characteristics of nonholonomic wheeled mobile robot including kinematical equation of motion is presented. And finally a spacecraft testbed model called *JC2Sat* is discussed.

2.1 Description and Dynamics of the Quadrotor UAV System

The Qball-X4 testbed is developed by Quanser Inc. through an NSERC-SPG project led by Concordia University (Fig. 2.1). It is enclosed within a protective carbon fiber round cage (therefore a name of Qball-X4) to ensure safe operation of the vehicle and protection to the personnel who is working with the vehicle in an indoor research and development environment. It uses four 10-inch propellers and standard motors and speed controllers. It is equipped with the Quanser Embedded Control Module (QECM), which is comprised of a Quanser HiQ aero data acquisition card and a QuaRC-powered Gumstix single-board embedded computer where QuaRC is Quanser's Real-time Control software. The Quanser

HiQ provides high-resolution accelerometer, gyroscope, and magnetometer IMU sensors as well as servo outputs to drive four motors. The on-board Gumstix computer runs QuaRC, which allows to rapidly develop and deploy controllers designed in MATLAB/Simulink environment to real-time control the Qball-X4. The controllers run on-board the vehicle itself and runtime sensors measurement, data logging and parameter tuning are supported between the ground host computer and the target vehicle [92].

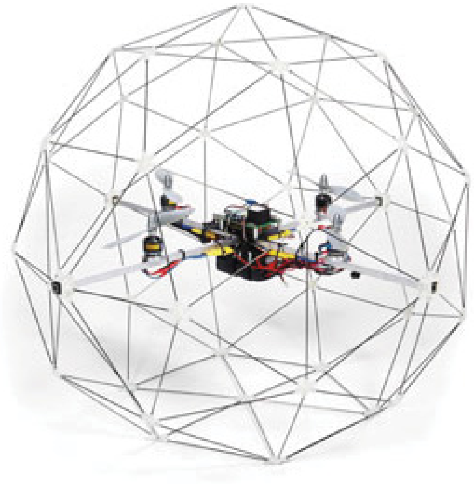


Figure 2.1: The Quanser Qball-X4 quadrotor UAV.

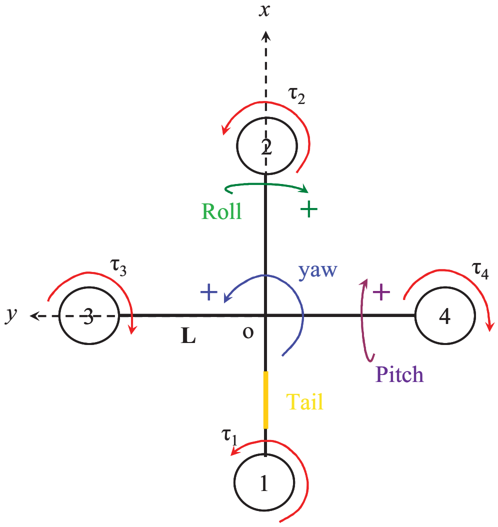


Figure 2.2: The Quanser Qball-X4 quadrotor UAV and its schematic representation.

The block diagram of the entire UAV system is illustrated in Figure 2.3. It is composed of three main parts:

- The first part represents the Electronic Speed Controllers (ESCs), the motors and the propellers in a set of four. The input to this part is $u = [u_1 \ u_2 \ u_3 \ u_4]^T$ which are Pulse Width Modulation (PWM) signals. The output is the thrust vector $T = [T_1 \ T_2 \ T_3 \ T_4]^T$ generated by four individually-controlled motor-driven propellers.
- The second part is the geometry that relates the generated thrusts to the applied lift and torques to the system. This geometry corresponds to the position and orientation of the propellers with respect to the system's center of mass.
- The third part is the dynamics that relate the applied lift and torques to the position P, velocity V and acceleration A of the Qball-X4.

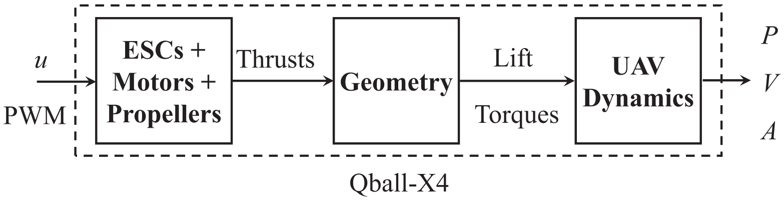


Figure 2.3: The UAV system block diagram.

The subsequent sections describe the corresponding mathematical model for each of the blocks in Figure 2.3.

2.1.1 Qball-X4 Dynamics

The Qball-X4 dynamics in a hybrid coordinate system are given hereafter where the position dynamics are expressed in the inertial frame and the angular dynamics are expressed in the body frame [93]:

$$
\begin{aligned}
m\ddot{x} &= u_z\,(cos\phi\ sin\theta\ cos\psi + sin\phi\ sin\psi) - k_x\dot{x} \\
m\ddot{y} &= u_z\,(cos\phi\ sin\theta\ sin\psi - sin\phi\ cos\psi) - k_y\dot{y} \\
m\ddot{z} &= u_z\,(cos\phi\ cos\theta) - mg - k_z\dot{z} \\
J_x\dot{p} &= u_p + (J_y - J_z)\,qr - J_T q\,\Omega - k_p p \\
J_y\dot{q} &= u_q + (J_z - J_x)\,pr - J_T p\,\Omega - k_q q \\
J_z\dot{r} &= u_r + (J_x - J_y)\,pq - k_r r
\end{aligned}
\tag{2.1}
$$

where x, y and z are the coordinates of the quadrotor UAV center of mass in the inertial frame. m is the system mass and J_x, J_y and J_z are the moments of inertia along y, x and z directions respectively. θ, ϕ and ψ are the pitch, roll and yaw Euler angles and p, q and r are the angular velocities in the body-fixed frame. k_x, k_y, k_z, k_p, k_q and k_r are drag coefficients and are constant. J_T is the moment of inertia for each motor and Ω is the overall speed of propellers:

$$\Omega = -\Omega_1 - \Omega_2 + \Omega_3 + \Omega_4 \tag{2.2}$$

where Ω_i is the i^{th} propeller speed.

The angular velocities in the inertial frame (Euler rates) can be related to those in the body frame as follows:

$$
\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & cos\theta sin\phi \\ 0 & -sin\phi & cos\theta cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.3}
$$

Close to hovering conditions, the matrix in the above equation is close to identity matrix and therefore the angular velocities in the body frame can be seen as the angular velocities in the inertial frame. The model (2.1) can then be written as:

$$\begin{aligned}
m\ddot{x} &= u_z\,(cos\phi\ sin\theta\ cos\psi + sin\phi\ sin\psi) - k_x\dot{x}\\
m\ddot{y} &= u_z\,(cos\phi\ sin\theta\ sin\psi - sin\phi\ cos\psi) - k_y\dot{y}\\
m\ddot{z} &= u_z\,(cos\phi\ cos\theta) - mg - k_z\dot{z}\\
J_x\ddot{\theta} &= u_\theta + (J_y - J_z)\,\dot{\phi}\dot{\psi} - J_T\dot{\phi}\ \Omega - k_\theta\dot{\theta}\\
J_y\ddot{\phi} &= u_\phi + (J_z - J_x)\,\dot{\theta}\dot{\psi} - J_T\dot{\theta}\ \Omega - k_\phi\dot{\phi}\\
J_z\ddot{\psi} &= u_\psi + (J_x - J_y)\,\dot{\theta}\dot{\phi} - k_\psi\dot{\psi}
\end{aligned} \tag{2.4}$$

where u_p, u_q, u_r, k_p, k_q and k_r have been respectively changed to u_θ, u_ϕ, u_ψ, k_θ, k_ϕ, k_ψ for notation convenience. At low speeds, one can obtain a simplified nonlinear model of (2.4) by neglecting drag terms and gyroscopic and Coriolis-centripetal effects:

$$\begin{aligned}
m\ddot{x} &= u_z\,(cos\phi\ sin\theta\ cos\psi + sin\phi\ sin\psi)\\
m\ddot{y} &= u_z\,(cos\phi\ sin\theta\ sin\psi - sin\phi\ cos\psi)\\
m\ddot{z} &= u_z\,(cos\phi\ cos\theta) - mg\\
J_x\ddot{\theta} &= u_\theta\\
J_y\ddot{\phi} &= u_\phi\\
J_z\ddot{\psi} &= u_\psi
\end{aligned} \tag{2.5}$$

A further simplified linear model can be obtained by assuming hovering conditions ($u_z \approx mg$ in the x and y directions) with no yawing ($\psi = 0$) and small roll and pitch angles:

$$\begin{aligned}
\ddot{x} &= \theta g; & J_x\ddot{\theta} &= u_\theta\\
\ddot{y} &= -\phi g; & J_y\ddot{\phi} &= u_\phi\\
\ddot{z} &= u_z/m - g; & J_z\ddot{\psi} &= u_\psi
\end{aligned} \tag{2.6}$$

2.1.2 ESCs, Motors and Propellers

The motors of the Qball-X4 are outrunner brushless motors. The generated thrust T_i of the i^{th} motor is related to the i^{th} PWM input u_i by a first-order linear transfer function:

$$T_i = K\frac{\omega}{s+\omega}u_i \;\; ; \;\; i = 1, \ldots, 4 \tag{2.7}$$

where K is a positive gain and ω is the motor bandwidth. K and ω are theoretically the same for the four motors but this may not be the case in practice and therefore, this can be one of sources of modeling errors/uncertainties for the FDD schemes.

2.1.3 Geometry

A schematic representation of the Qball-X4 is given in Figure 2.2. The motors and propellers are configured in such a way that the back and front (1 and 2) motors spin clockwise and the left and right (3 and 4) motors spin counter-clockwise. Each motor is located at a distance L from the center of mass o and when spinning, a motor produces a torque τ_i. The origin of the body-fixed frame is the system's center of mass o with the x-axis pointing from back to front and the y-axis pointing from right to left. The thrust T_i generated by the i^{th} propeller is always pointing upward in the z-direction in parallel to the motor's rotation axis. The thrusts T_i and the torques τ_i result in a lift in the z-direction (body-fixed frame) and torques about the x, y and z axes. The relations between the lift/torques and the thrusts are:

$$\begin{aligned}
u_z &= T_1 + T_2 + T_3 + T_4 \\
u_\theta &= L(T_1 - T_2) \\
u_\phi &= L(T_3 - T_4) \\
u_\psi &= \tau_1 + \tau_2 - \tau_3 - \tau_4
\end{aligned} \tag{2.8}$$

The torque τ_i produced by the i^{th} motor is directly related to the thrust T_i via the relation of $\tau_i = K_\psi T_i$ with K_ψ as a constant. In addition, by setting $T_i \approx K u_i$ from (2.7), the relation (2.8) reads:

$$\begin{aligned}
u_z &= K(u_1 + u_2 + u_3 + u_4) \\
u_\theta &= KL(u_1 - u_2) \\
u_\phi &= KL(u_3 - u_4) \\
u_\psi &= KK_\psi(u_1 + u_2 - u_3 - u_4)
\end{aligned} \tag{2.9}$$

where u_z is the total lift generated by the four propellers and applied to the quadrotor UAV in the z-direction (body-fixed frame). u_θ, u_ϕ and u_ψ are respectively the applied torques in θ, ϕ and ψ directions (see Figure 2.2).

2.2 Description and Dynamics of Wheeled Mobile Robot

In this section, kinematical model of mobile robot is described with consideration of the nonholonomic constraint. To simulate the robot behavior, an exact mathematical model is required. A complete study of the kinematical model of WMRs could be found in [94]. In the current study, a three-wheeled mobile robot is considered as depicted in Fig. 2.4. The platform is differentially-driven, so the front wheels are active and independent, hence performing both the driving and the steering of the system. The other wheel, point 'C' as shown in Fig. 2.4, is a caster and just adds stability to the system.

In order to describe the platform position, the middle point of the rear axle, point 'G', is considered as a reference point. The coordinates of this point are represented as $(X(t), Y(t))$. The angle between the longitudinal axis of the robot and the horizontal axis is called heading angle and denoted by $\theta(t)$, which is assumed to be in the range $-\pi \leq \theta \leq \pi$. Besides, $v(t)$ and $\omega(t)$, in Fig. 2.4, represent the linear and angular velocities of the robot, respectively. Moreover, the 'XY' coordinate system denotes an inertial frame of reference, whereas 'xy' represents the moving and rotating coordinate frame attached to the platform at point ‘*G*’.

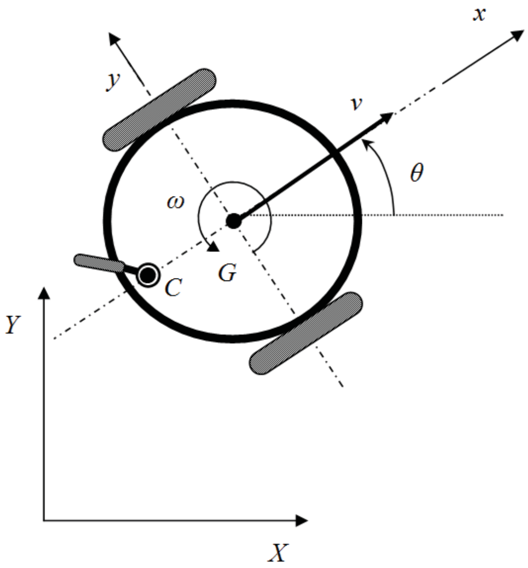


Figure 2.4: The schematic of the mobile robot.

The governing kinematical equations of motion of the robot are written as (2.10):

$$\begin{aligned}\dot{X} &= v(t)cos(\theta(t))\\ \dot{Y} &= v(t)sin(\theta(t))\\ \dot{\theta} &= \omega(t)\end{aligned} \tag{2.10}$$

If the reference trajectory of the robot is considered as $(X_r(t), Y_r(t))$, then other kinematical characteristics of reference trajectory can be obtained as (2.11-2.13):

$$v_r(t) = \pm\sqrt{\dot{X}_r(t)^2 + \dot{Y}_r(t)^2} \tag{2.11}$$

$$\omega_r(t) = \frac{\ddot{Y}_r(t)\dot{X}_r(t) - \ddot{X}_r(t)\dot{Y}_r(t)}{\dot{X}_r(t)^2 + \dot{Y}_r(t)^2} \tag{2.12}$$

$$\theta_r(t) = atan2(\dot{Y}_r(t), \dot{X}_r(t)) \tag{2.13}$$

where $v_r(t)$, $\omega_r(t)$, $\theta_r(t)$ are reference linear velocity, reference angular velocity, and reference heading angle of the robot respectively. In (2.11), the '+' sign corresponds to the forward motion while the '-' sign is used for backward motion.

2.2.1 Qbot

As a testbed for wheeled mobile robot's related applications a robotic vehicles called "Qbot" is used (Fig. 2.5). Qbot ia a multi-purpose research robotic system developed by Quanser Inc. It has two differentially driven wheels, and a caster wheel (to maintain its stability).

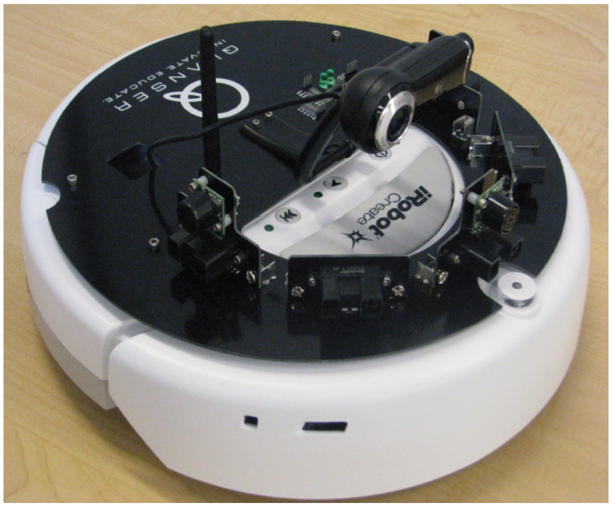


Figure 2.5: Qbot, unmanned ground vehicle developed by Quanser Inc.

2.3 JC2Sat-FF Mission and JC2Sat Attitude Model

2.3.1 JC2Sat Formation Flying Mission

The JC2Sat-FF is a joint Canadian Space Agency/Japan Aerospace Exploration Agency technology mission consisting of two nano-satellites and weighing approximately at 18kg each. The main objective of this mission is to demonstrate the feasibility of maintaining along-track spacecraft formation by using only differential atmospheric drag control between the two satellites and GPS-based relative navigation (Fig. 2.6). During nominal mission operations, the satellites are three-axis stabilized with the pitch axis of each satellite body frame aligned with the orbit normal [1]. The differential drag between the two satellites is controlled by simply varying the pitch angle of each satellite, which results in a change of the frontal drag area. The pitch manoeuvre is provided by the attitude control system which uses magnetic torque rods and momentum wheels as actuators.

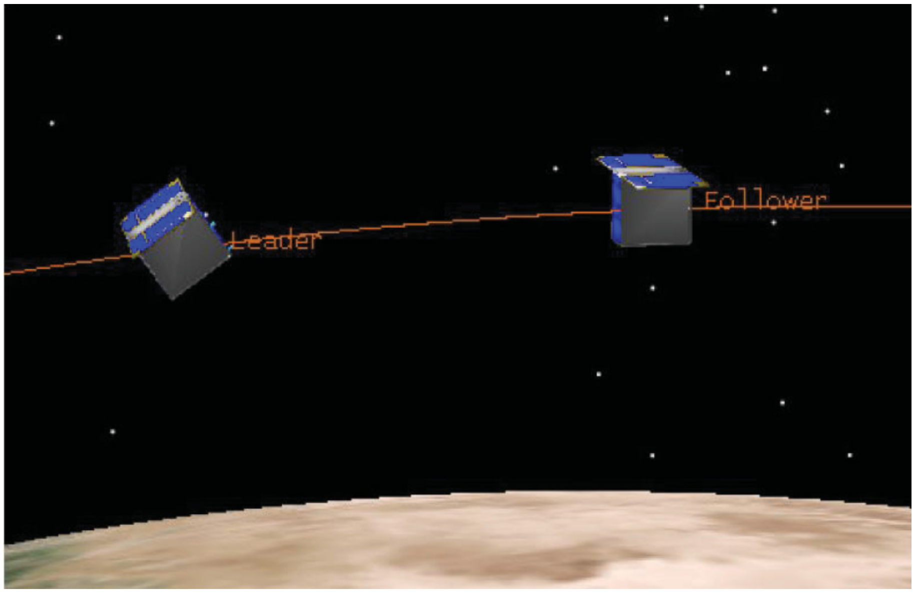


Figure 2.6: JC2Sat-FF mission concept [1].

The sensors and actuators of the Attitude Control System (ACS) of each satellite include two medium accuracy digital sun sensors, one three-axis magnetometer, three magnetic torque rods and two momentum wheels with spinning axes aligned with the pitch axis of the satellite. In nominal mission operations, the magnetorquer provide roll and yaw control and momentum wheel de-saturation, while the two momentum wheels provide the bias momentum about satellite pitch axis and the fast pitch control [64]. Figure 2.7 demonstrates mission stages with respect to satellite configurations. The two satellites will be launched together in stack configuration. Prior to inter-satellite separation, only ACS of the leader satellite is active which detects and controls the attitude of the satellite stack. Inter-satellite separation has special requirements on stack attitude to prevent loss of formation and to avoid collision of the two satellites. In preparing for the separation, the satellite stack is maneuvered to track a predefined separation frame whose pitch axis is inertially fixed in space. At the point of separation, the satellite stack is nadir pointing with body pitch axis pointing 10 degrees away from the orbit normal. The onboard actuators of the satellites are basically single string with little redundancy. Therefore, their reliability and availability are very important for the success of the mission.

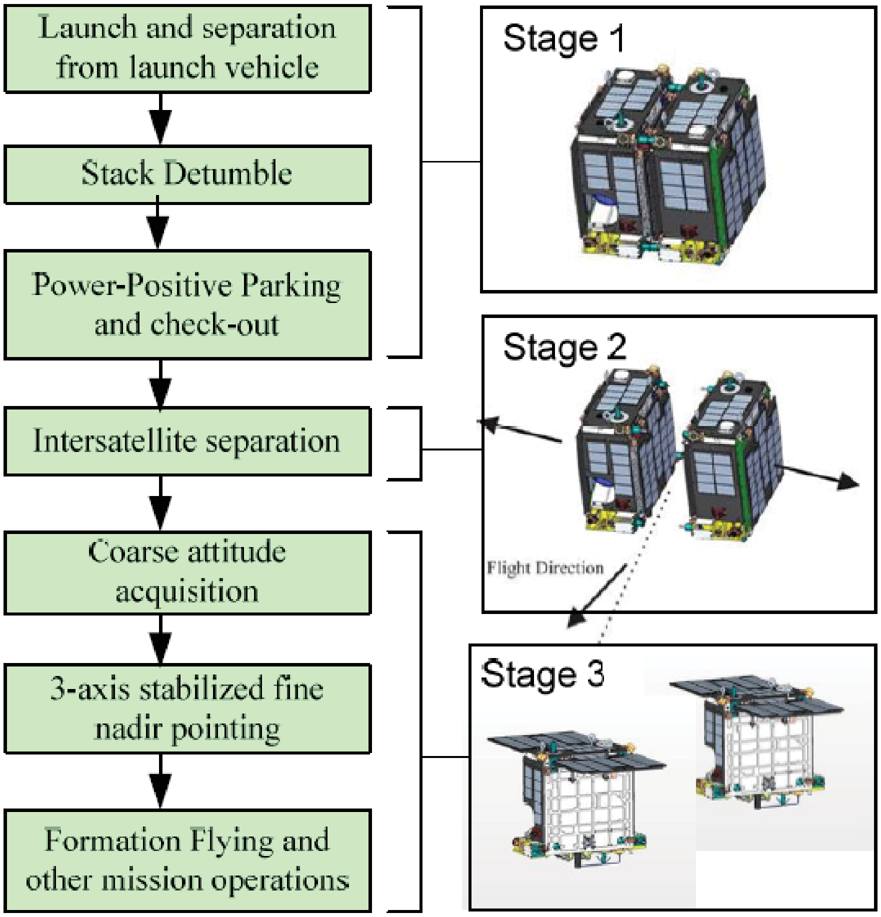

Launch and separation from launch vehicle
Stack Detumble
Power-Positive Parking and check-out
Intersatellite separation
Coarse attitude acquisition
3-axis stabilized fine nadir pointing
Formation Flying and other mission operations
Stage 1
Stage 2
Flight Direction
Stage 3

Figure 2.7: Mission stages with respect to satellite configurations [1].

In nominal mission operations, the magnetorquers provide roll and yaw control and momentum wheel de-saturation, while the two momentum wheels provide the bias momentum about satellite pitch axis and the fast pitch control [1]. The momentum wheels also play an important role during the inter-satellite separation of the two satellites (which are tied together during launch) by providing bias momentum about the stack pitch axis and the pitch manoeuvre.

2.3.2 JC2Sat Attitude Model

In this subsection nonlinear attitude model is presented based on the reference frames presented in Fig. 2.8. Based on the Newton-Eulers moment equation the spacecraft dynamics can be written as (2.14):

$$\mathbf{M} = \dot{\mathbf{H}}_\mathbf{I} = \dot{\mathbf{H}}_\mathbf{B} + \omega \mathbf{H} \tag{2.14}$$

where $\mathbf{M}$ is the total external torque, $\mathbf{H_I}$ is the angular momentum in inertial reference frame, $\mathbf{H_B}$ is angular moment in body reference frame, and ω is angular velocity of the satellite. Thus the complete nonlinear dynamic model of the spacecraft attitude is given in Eq. (2.15) as follows:

$$\begin{aligned} \mathbf{M_x} &= \dot{\mathbf{H}}_\mathbf{x} + \mathbf{H_z}\omega_\mathbf{y} - \mathbf{H_y}\omega_\mathbf{z} \\ \mathbf{M_y} &= \dot{\mathbf{H}}_\mathbf{y} + \mathbf{H_x}\omega_\mathbf{z} - \mathbf{H_z}\omega_\mathbf{x} \\ \mathbf{M_z} &= \dot{\mathbf{H}}_\mathbf{z} + \mathbf{H_y}\omega_\mathbf{x} - \mathbf{H_x}\omega_\mathbf{y} \end{aligned} \tag{2.15}$$

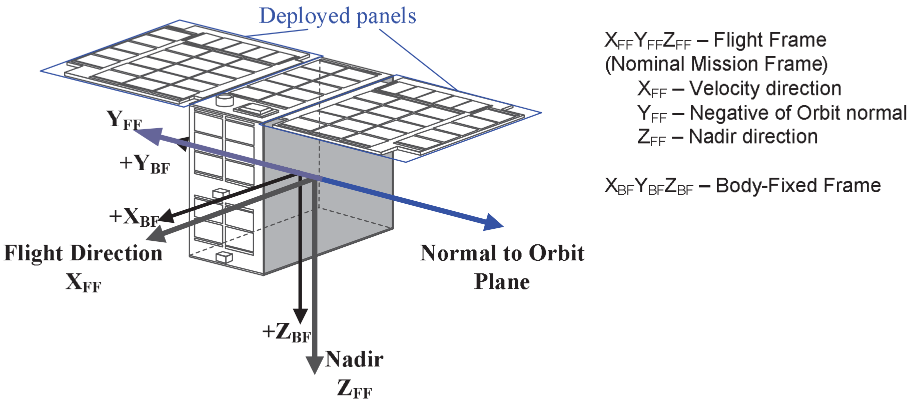

Deployed panels
Y_{FF}
$+Y_{BF}$
$+X_{BF}$
Flight Direction X_{FF}
Normal to Orbit Plane
$+Z_{BF}$
Nadir Z_{FF}
$X_{FF}Y_{FF}Z_{FF}$ – Flight Frame (Nominal Mission Frame)
X_{FF} – Velocity direction
Y_{FF} – Negative of Orbit normal
Z_{FF} – Nadir direction
$X_{BF}Y_{BF}Z_{BF}$ – Body-Fixed Frame

Figure 2.8: Reference frames and nominal attitude of the JC2Sat-FF satellite.

In the JC2Sat, magnetorquers provide roll and yaw control and momentum wheel desaturation. Pitch control is provided by the momentum wheel (spinning axes of the wheels are aligned with satellite body pitch axis). During a coarse earth pointing operation mode, or so-called acquisition mode [1], only one of the two momentum wheels is in operation at low spin speed. In this mode, the satellite is controlled as a non-gyroscopic rigid body. A Proportional-Derivative (PD) control law for each axis is used to achieve three-axis control. The FDD algorithm will be designed for this operation mode.

Let φ, θ, and ψ be the roll, pitch and yaw angles describing the rotation from the flight frame to the satellite body-fixed control frame as shown in Fig. 2.8. The control laws in attitude control system for each axis are summarized in Eq. (3). T_{cx}, T_{cy}, T_{cz} represent control torque in each axis and K_p, K_d represent proportional and derivative controller gains. As shown in Eq. (2.16), for pitch axis the control torque is obtained from the momentum wheel $T_{cy} = T_{MW}$. More detailed information can be found in [1].

$$\begin{aligned} T_{cx} &= -K_{px}\varphi - K_{dx}\dot{\varphi} \\ T_{cy} = T_{MW} &= -K_{py}\theta - K_{dy}\dot{\theta} \\ T_{cz} &= -K_{pz}\psi - K_{dz}\dot{\psi} \end{aligned} \tag{2.16}$$

Given the spacecraft attitude model along with the controller the objective of the next section is to design FDD algorithm for controller in the closed-loop momentum wheel actuator faults detection and diagnosis.

2.3.3 Dynamic Model of JC2Sat's Momentum Wheel

The actuator which is the momentum wheel (Fig 2.9) used in JC2Sat-FF is manufactured by Sinclair Interplanetary in collaboration with the Space Flight Laboratory of University of Toronto [95]. Two identical momentum wheels are used. Each wheel provides 0.05 Nm-sec bias momentum with 0.5 W power consumption. The spin axes of both wheels are aligned with the pitch axis of the satellite. The momentum wheel is specifically designed and manufactured for spacecraft in the 2-20 kg range. The main benefits of this particular momentum wheel include scalability and low-cost by virtue of a custom motor that does not require a pressurized container. The specifications of the momentum wheel can be found in [95], as given in Table 2.1.

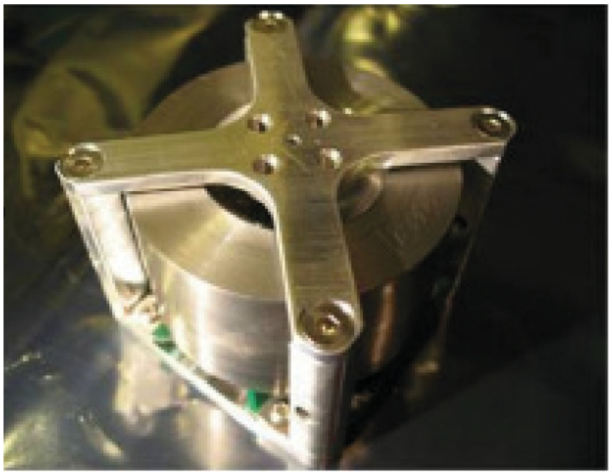


Figure 2.9: JC2Sat-FF momentum wheel [1].

Fig. 2.10 shows a detailed block diagram of momentum wheel used for simulation purposes. In this model V is the input voltage of the electrical motor, R_M is the electrical resistance of the motor armature, and K_M is the torque coefficient of the motor. B is friction coefficient and K_{BEMF} is back electromotive force coefficient. The term I_w denotes the overall moment of inertia of the rotor including the flywheel, and I_s is the moment of inertia of the satellite.

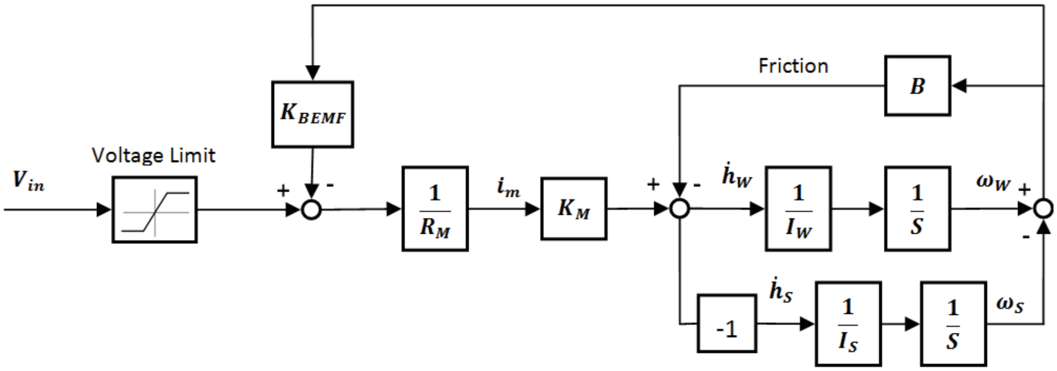


Figure 2.10: A detailed block diagram of momentum wheel.

The primary task of the electrical motor is to provide the necessary angular torque to the satellite. According to Euler's moment equation of angular motion, if there are no

Table 2.1: Specifications of momentum wheel used in JC2Sat-FF mission

Product number	RQ-0.06-28-ASYNC-2-1-x
Mass	25g
Dimensions	65mm x 65mm x 37.5 mm
Power consumption	2W at full torque, 0.5 W @ 5000 RPM, 0.2 W @ 2000 RPM
Operating voltage	7.5V to 35 V
Rotor imbalance force	< 130mN @ 5000RPM
Rotor imbalance torque	< 3mNm @ 5000RPM
Angular momentum	60mNm-sec
Maximum speed	> 8000RPM
Communication	Asynchronous serial
Bearings	Hybrid/Diamond ABEC7
Redundancy	Dual wound
Radiation	Standard parts
Operating temperature	-40c to 70c
Control mode	Speed or Torque, built-in control CPU

external disturbances or inertial control torques acting on the satellite then $\dot{h}_w + \dot{h}_s = 0$. This means that, in order to apply a torque on the body about an axis, a torque in the opposite direction must be produced by the rotor of the electrical motor. Designing a model-based FDD method involves the mathematical model of the momentum wheel. The linearized equation of momentum wheel can be written in the state-space form as shown in Eq. 2.17. It is assumed that the states are angular velocity of the wheel and angular velocity of the spacecraft. The outputs are the angular velocities of spacecraft along with motor current (Eq. 2.18).

$$\begin{bmatrix} \dot{\omega}_s \\ \dot{\omega}_w \end{bmatrix} = \begin{bmatrix} -\frac{K_M K_{BEMF}}{R_M I_s} - \frac{B}{I_s} & \frac{K_M K_{BEMF}}{R_M I_s} + \frac{B}{I_s} \\ \frac{K_M K_{BEMF}}{R_M I_w} + \frac{B}{I_w} & -\frac{K_M K_{BEMF}}{R_M I_w} - \frac{B}{I_w} \end{bmatrix} \begin{bmatrix} \omega_s \\ \omega_w \end{bmatrix} + \begin{bmatrix} \frac{K_M}{R_M I_s} \\ \frac{K_M}{R_M I_w} \end{bmatrix} V_{in} \qquad (2.17)$$

$$Z = \begin{bmatrix} \omega_s \\ i \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \\ \dfrac{K_{BEMF}}{R_M} & -\dfrac{K_{BEMF}}{R_M} \end{bmatrix} \begin{bmatrix} \omega_s \\ \omega_w \end{bmatrix} + \begin{bmatrix} 0 \\ \\ \dfrac{1}{R_M} \end{bmatrix} V_{in} \tag{2.18}$$

2.4 Chapter Summary

In this chapter, dynamics and mathematical models of three types of testbeds were described. Description and dynamics of *Qball-X4* were presented firstly. The characteristics of nonholonomic wheeled mobile robot including kinematical equation of motion was presented secondly. Finally *JC2Sat* space mission was discussed. Next chapter will present some Fault Detection and Diagnosis (FDD) methods for application to Unmanned Aerial Vehicles (UAV) and Spacecrafts.

Chapter 3

Fault Detection and Diagnosis Algorithms

This chapter is dedicated to description of three different FDD methods which have been used in this thesis for the purpose of fault detection, isolation and identification. Firstly, a Two-Stage Kalman Filter is presented. Then Adaptive Two-Stage Kalman Filter is explained and finally an Interacting Multiple Model is presented.

3.1 Two-Stage Kalman Filter

The advantage of using Two-Stage Kalman Filter (TSKF) is to simultaneously estimate both states and fault parameters, for the purpose of fault detection, isolation and identification as well as providing full state estimation for state feedback-based controllers when state vector is not available through measurements. To explain the basic idea of the TSKF the following discrete linear state-space model is considered:

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k + v_k^x \\ y_{k+1} &= C_k x_{k+1} + v_{k+1}^y \end{aligned} \tag{3.1}$$

where $x_k \in R^n$, $u_k \in R^l$ and $y_{k+1} \in R^m$ are the state, control input and output vectors respectively. v_k^x and v_{k+1}^y are uncorrelated Gaussian random vectors with zero means and covariance matrices Q_k^x and R_k, respectively. In the application of Kalman filtering techniques, an accurate model of the process dynamics and measurements is required. However, in many practical cases, constant bias affects the system dynamics and observations, and may lead to performance degradation of the filter if the bias is not incorporated in the model. By considering a bias vector $\gamma_k \in R^l$ in the state-space equation, model (3.1) can be written as:

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k + F\gamma_k + v_k^x \\ \gamma_{k+1} &= \gamma_k + v_k^\gamma \\ y_{k+1} &= C_k x_{k+1} + v_{k+1}^y \end{aligned} \tag{3.2}$$

where v_k^γ is an uncorrelated Gaussian random vector with zero mean and covariance matrix Q_k^γ. One possible solution to estimate the bias vector γ_k is to augment it into the state vector to make an augmented state vector which is estimated by using the Augmented State Kalman Filter (ASKF). The augmented state vector is of dimension $n+l$ which makes the ASKF computationally expensive. Another drawback of this method is that numerical problems may arise during implementation. To reduce the complexity of this problem, Keller and Darouach [68] presented two parallel reduced-order filters which optimally implement the augmented state filter. The proposed algorithm is called Two-Stage Kalman Filter (TSKF). In the context of fault detection and diagnosis, Wu *et al.* [69, 71] modeled actuator faults (loss of control effectiveness) as a bias vector in state equations then used TSKF to estimate the bias vector. In [69], the authors proposed the use of forgetting factor for TSKF.

In the current study, the effectiveness of actuators is estimated as the random bias vector in the TSKF structure. By referring to (3.2), the loss of control effectiveness modeled as a bias vector is:

$$\gamma_k = \begin{bmatrix} \gamma_{1_k} & \gamma_{2_k} & \dots & \gamma_{l_k} \end{bmatrix}^T \; ; \; 0 \le \gamma_{i_k} \le 1 \; ; \; i = 1, 2, \dots, l \tag{3.3}$$

where $\gamma_{i_k} = 0$ and $\gamma_{i_k} = 1$ means that the i^{th} actuator is completely healthy or fully damaged, respectively. By incorporating actuator faults in (3.2), the bias augmented discrete linear time-varying state-space model is written as:

$$x_{k+1} = A_k x_k + B_k u_k - B_k U_k \gamma_k + v_k^x \tag{3.4}$$

$$\gamma_{k+1} = \gamma_k + v_k^\gamma \tag{3.5}$$

$$y_{k+1} = C_k x_{k+1} + v_{k+1}^y \tag{3.6}$$

where U_k is:

$$U_k = \begin{bmatrix} u_{1_k} & 0 & \dots & 0 \\ 0 & u_{2_k} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & u_{l_k} \end{bmatrix} \tag{3.7}$$

Since the additive noise v_k^γ introduced into the bias evolution equation bears no relation to either the process noise v_k^x or the measurement noise v_{k+1}^y in the dynamic system model, the two-stage filtering algorithm by Keller and Darouach can be applied with some modification to obtain the bias estimates [71]. It's worthy to note that the choice of Q_k^γ plays an important role in the performance of the filter. In the most of real world situations the loss of control effectiveness occurs abruptly while in (3.4) it is modeled as a bias. To make the bias vector change fast enough (and also to track the true values) one may think of a large value for bias covariance. On the other hand once the steady-state condition is reached, the bias covariance must be small to reduce the estimation steady-state error. Then there is a trade-off between convergence speed and the accuracy of steady-state estimation of control effectiveness. The minimum variance solution to estimate the true values of biases is obtained by application of the TSKF.

Like other types of Kalman filters, the equations can be divided into two sets: the *time-update* equations and the *measurement-update* equations. The time-update equations which can be distinguished by the $(k+1|k)$ subscription, are responsible to obtain *a priori* estimates by moving the state and error covariances one step ahead in the time domain. The measurement-update equations, shown by the $(k+1|k+1)$ subscription, are responsible to obtain *a posteriori* estimates through feed-backing measurements into the *a priori* estimates. In other words time-update equations are used for prediction, while measurement-update equations are used for correction. Indeed the whole prediction-correction process is used to estimate the states as close as possible to their real values. Figure 3.1 shows a schematic flow diagram of the TSKF and its implementation is given hereafter.

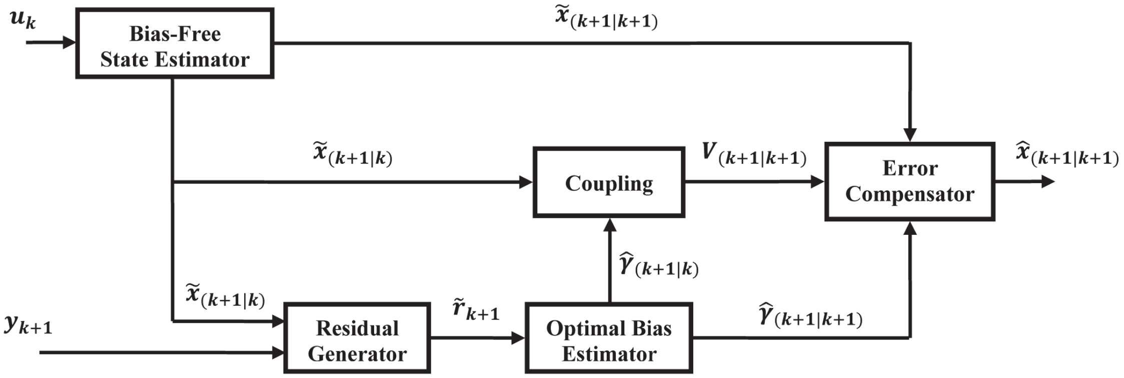


Figure 3.1: The schematic diagram of the Two-Stage Kalman Filter (TSKF).

Estimate the bias-free state:

$$\tilde{x}_{k+1|k} = A_k \tilde{x}_{k|k} + B_k u_k + W_k \hat{\gamma}_{k|k} - V_{k+1|k} \hat{\gamma}_{k|k} \tag{3.8}$$

$$\tilde{P}^x_{k+1|k} = A_k \tilde{P}^x_{k|k} A_k^T + Q^x_k + W_k P^\gamma_{k|k} W_k^T - V_{k+1|k} P^\gamma_{k+1|k} V^T_{k+1|k} \tag{3.9}$$

$$\tilde{x}_{k+1|k+1} = \tilde{x}_{k+1|k} + \tilde{K}^x_{k+1}(y_{k+1} - C_{k+1}\tilde{x}_{k+1|k}) \tag{3.10}$$

$$\tilde{K}^x_{k+1} = \tilde{P}^x_{k+1|k} + C^T_{k+1}(C_{k+1}\tilde{P}^x_{k+1|k}C^T_{k+1} + R_{k+1})^{-1} \tag{3.11}$$

$$\tilde{P}^x_{k+1|k+1} = (I - \tilde{K}^x_{k+1}C_{k+1})\tilde{P}^x_{k+1|k} \tag{3.12}$$

Determine the filter residual and its covariance:

$$\tilde{r}_{k+1} = y_{k+1} - C_{k+1}\tilde{x}_{k+1|k} \tag{3.13}$$

$$\tilde{S}_{k+1} = C_{k+1}\tilde{P}^x_{k+1|k}C^T_{k+1} + R_{k+1} \tag{3.14}$$

Estimate the optimal bias:

$$\hat{\gamma}_{k+1|k} = \hat{\gamma}_{k|k} \tag{3.15}$$

$$P^\gamma_{k+1|k} = P^\gamma_{k|k} + Q^\gamma_k \tag{3.16}$$

$$\hat{\gamma}_{k+1|k+1} = \hat{\gamma}_{k+1|k} + K^\gamma_{k+1}(\tilde{r}_{k+1} - H_{k+1|k}\hat{\gamma}_{k|k}) \tag{3.17}$$

$$K^\gamma_{k+1} = P^\gamma_{k+1|k}H^T_{k+1|k}(H_{k+1|k}P^\gamma_{k+1|k}H^T_{k+1|k} + \tilde{S}_{k+1})^{-1} \tag{3.18}$$

$$P^\gamma_{k+1|k+1} = (I - K^\gamma_{k+1}H_{k+1|k})P^\gamma_{k+1|k} \tag{3.19}$$

Calculate the coupling equations:

$$W_k = A_kV_{k|k} - B_kU_k \tag{3.20}$$

$$V_{k+1|k} = W_kP^\gamma_{k|k}(P^\gamma_{k+1|k})^{-1} \tag{3.21}$$

$$H_{k+1|k} = C_{k+1}V_{k+1|k} \tag{3.22}$$

$$V_{k+1|k+1} = V_{k+1|k} - \tilde{K}^x_{k+1}H_{k+1|k} \tag{3.23}$$

Estimate the compensated error and its covariance:

$$\hat{x}_{k+1|k+1} = \tilde{x}_{k+1|k+1} + V_{k+1|k+1}\hat{\gamma}_{k+1|k+1} \tag{3.24}$$

$$P_{k+1|k+1} = \tilde{P}^{x}_{k+1|k+1} + V_{k+1|k+1}P^{\gamma}_{k+1|k+1} + V^{T}_{k+1|k+1} \tag{3.25}$$

3.2 Adaptive Two-Stage Kalman Filter

The Adaptive Two-Stage Kalman Filter (ATSKF) is addressed for the purpose of adaptive estimation of control effectiveness factors. To make the ordinary TSKF more responsive to abrupt changes in control effectiveness factor (which has been modeled as a bias vector in (3.4)) some modification has been suggested in [71]. A common way to make recursive algorithm (like Kalman filter's family) less sensitive to previous data history is to use forgetting factor. In other words, the main goal of using forgetting factor is to make the Kalman filter more sensitive to abrupt changes in the control effectiveness parameters. When a forgetting factor is added to the optimal bias estimator part in (3.16), the new time updated equation of bias covariance is written as:

$$P^{\gamma}_{k+1|k} = P^{\gamma}_{k|k}/\lambda + Q^{\gamma}_{k} \quad , \quad 0 < \lambda \leq 1 \tag{3.26}$$

where λ is a forgetting factor which can have a predefined constant value. Using a constant value for forgetting factor may lead to instability of the TSKF. Indeed estimation error covariance matrix may increase rapidly due to the recursive nature of the algorithm. One way to prevent "blow up" in TSKF is to keep $P^{\gamma}_{k+1|k}$ in a certain range. Parkum *et al.* [96] used a non-uniform forgetting factor in recursive least-square-based parameter identification schemes. The idea is that, under the normal system operation condition, the error covariance matrix $P^{\gamma}_{k|k}$ describes the bias estimation error. The bias estimation error should be kept in a range means that the error covariance matrix should not be so large or so small.

Inspired by this fact, an adaptive technique is suggested in [96] and considered later in [70, 71]. This adaptive technique forces the error covariance matrix $P^{\gamma}_{k+1|k}$ to stay within pre-described limits:

$$\sigma_{min} I \leq P^{\gamma}_{k+1|k} \leq \sigma_{max} I \tag{3.27}$$

where σ_{min} and σ_{max} are positive constants and I is the identity matrix. A Dyadic expansion of $P^{\gamma}_{k|k}$ is given as:

$$P^{\gamma}_{k|k} = \sum_{i=1}^{l} \alpha^{i}_{k|k} e^{i}_{k} (e^{i}_{k})^{T} \quad , \quad \|e^{1}_{k}\| = \cdots = \|e^{l}_{k}\| = 1 \tag{3.28}$$

where $e^{1}_{k}, \ldots, e^{l}_{k}$ and $\alpha^{1}_{k}, \ldots, \alpha^{l}_{k}$ are the eigenvectors and eigenvalues of $P^{\gamma}_{k|k}$ respectively. Substituting (3.28) into (3.26) gives:

$$P^{\gamma}_{k+1|k} = \sum_{i=1}^{l} \frac{\alpha^{i}_{k|k}}{\lambda^{i}_{k}} e^{i}_{k} (e^{i}_{k})^{T} + Q^{\gamma}_{k} \quad , \quad 0 < \lambda^{i}_{k} \leq 1 \tag{3.29}$$

Replacing (3.16) by (3.29) the TSKF will turn to ATSKF. As suggested in [96] the forgetting factor λ^{i}_{k} can be chosen as a decreasing function of the amount of information received in the direction e^{i}_{k}. Since eigenvalue $\alpha^{i}_{k|k}$ of $P^{\gamma}_{k|k}$ is a measure of the uncertainty in the direction of e^{i}_{k}, a choice of forgetting factor λ^{i}_{k} based on the above constraints can be:

$$\lambda^{i}_{k} = \begin{cases} 1 & \text{if} \quad \alpha^{i}_{k|k} > \alpha_{max} \\ \alpha^{i}_{k|k} \left[\alpha_{min} + \dfrac{\alpha_{max} - \alpha_{min}}{\alpha_{max}} \alpha^{i}_{k|k} \right]^{-1} & \text{if} \quad \alpha^{i}_{k|k} \leq \alpha_{max} \end{cases} \tag{3.30}$$

where α_{max} and α_{min} are positive constants. The choice of α_{max} and α_{min} is made by the designer. If the ratio $\alpha_{max}/\alpha_{min}$ is close to one then the adaptive forgetting factor approaches a constant value.

3.3 Interacting Multiple Model Filter

The Interacting Multiple Model (IMM) method is one of the most efficient approaches for FDD applications, which was first published in [72]. The IMM runs a bank of filters in parallel, each based on model matching to a particular mode (healthy or faulty) of the system and by switching from one model to the other in a probabilistic manner. Each filter interacts with each other in a highly cost-effective fashion and thus leads to significantly improved performance. The initial estimate at the start of each cycle for each filter is a mixture of all most recent estimates from each filter. It is this mixing and interacting that offers advantages in IMM to effectively consider the small changes induced by fault quickly, which is mostly failed to be recognized by conventional multiple model approaches. Such a significant feature makes IMM approach much more suitable for FDD or manoeuvring target tracking applications. A summary of the IMM method is provided below and for a complete description of IMM the interested readers are referred to [72]. The IMM algorithm in each step (cycle) consists of four steps which are interacting/mixing, filtering, mode probability update and final combination of the models which provides the combined state estimate and its associated covariance matrix. In addition, for the above-mentioned four steps, fault isolation can also be performed based on the probability function. All the five steps are shown in Fig. 3.2.

The IMM-based FDD scheme assumes that the actual system at any time can be modelled sufficiently by a stochastic hybrid system given in (3.31).

$$\begin{aligned} x(k+1) &= F(k,m(k+1))x(k) + G(k,m(k+1))u(k) \\ &\quad + T(k,m(k+1))\varepsilon(k,m(k+1)) \\ z(k) &= H(k,m(k))x(k) + D(k,m(k))u(k) + \eta(k,m(k)) \end{aligned} \tag{3.31}$$

with the system mode sequence assumed to be a first-order Markov chain with following transition probabilities:

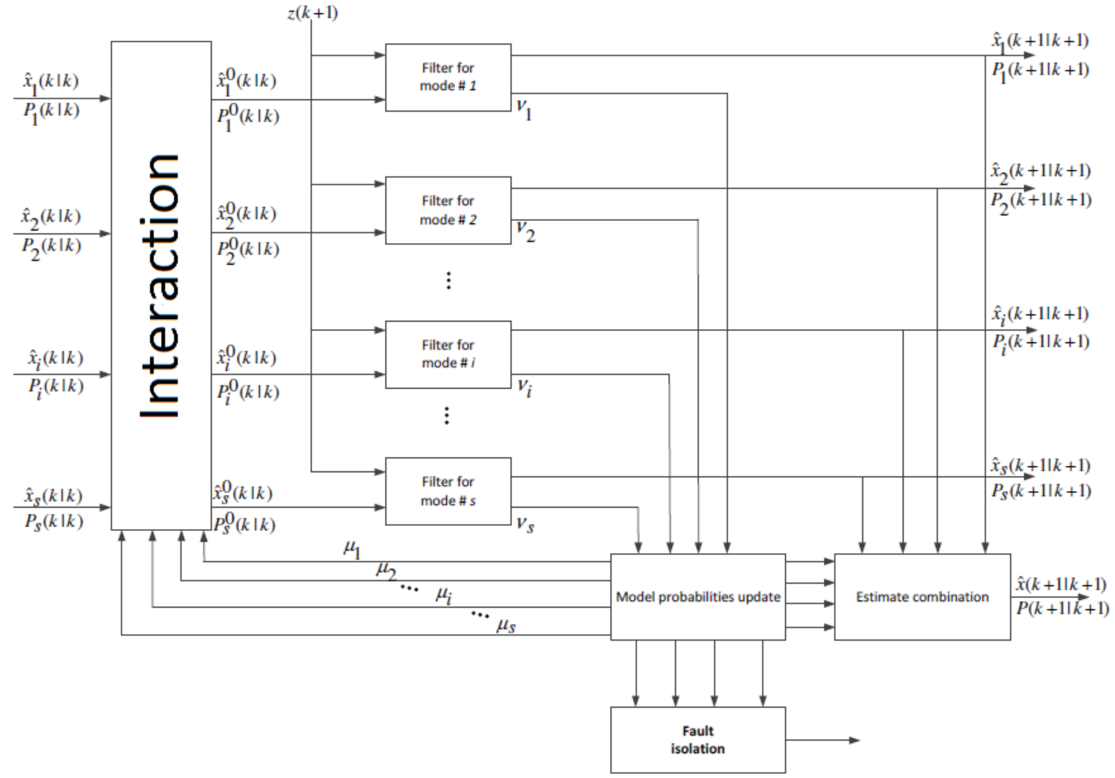


Figure 3.2: Block diagram of IMM-based FDD.

$$P\{m_j(k+1)|m_i(k)\} = \pi_{ij}(k) \quad \forall m_i, m_j \in S \tag{3.32}$$

and

$$\sum_j \pi_{ij}(k) = 1 \quad i = 1, \ldots, s \tag{3.33}$$

here $x \in R^{n_x}$ is the base state vector, $z \in R^{n_z}$ is the measurement vector, $u \in R^{n_u}$ is control input vector, $\varepsilon \in R^{n_n}$ and $\eta \in R^{n_z}$ are mutually independent discrete-time process and measurement noises with mean $\bar{\varepsilon}$ and $\bar{\eta}$, covariance $Q(k)$ and $R(k)$; $P\{.\}$denotes probability; $m(k)$ is the discrete-valued modal state at time k, which denotes the mode in effect during the sampling period ending at t_k; π_{ij} is the transition probability from mode m_i to mode m_j; the event that m_j is in effect at time k is denoted as $m_j(k) \triangleq \{m(k) = m_j\}$. $S = \{m_1, m_2, \ldots, m_s\}$ is the set of all possible system modes; the initial state is assumed to have mean $\hat{x}_0$ and covariance P_0 and be independent of ε and η .

The mathematical description of IMM-based FDD scheme is given in the following equations. As mentioned earlier, we will divide the IMM procedure for FDD into 4 different steps. The first involves interacting/mixing of the estimates for the four different models considered. The predicted mode probability and mixing probability at one cycle is given in Eq. (3.34).

$$\begin{aligned} \mu_j(k+1|k) &\triangleq P\{m_j(k+1)|z^k\} = \sum_{i=1}^{s} \pi_{ij}\mu_j(k) \\ \mu_{i|j}(k) &\triangleq P\{m_j(k)|m_j(k+1),z^k\} = \frac{\pi_{ij}\mu_i(k)}{\mu_j(k+1|k)} \qquad i,j=1,\ldots,s \end{aligned} \tag{3.34}$$

Based on the mixing probability and mode probability, the mixed initial state and covariance estimates to be used for next time-step of filter update are given in Eq. (3.35).

$$\begin{aligned} \hat{x}_j^0 &\triangleq E\left[x(k)|m_j(k+1),z^k\right] = \sum_{i=1}^{s} \hat{x}_i(k|k)\mu_{i|j}(k) \qquad j=1,\ldots,s \\ P_j^0(k|k) &\triangleq cov\left[\hat{x}_j^0|m_j(k+1),z^k\right] \\ &= \sum_{i=1}^{s}\left[P_i(k|k)\left[\hat{x}_j^0(k|k)-\hat{x}_j(k|k)\right]\left[\hat{x}_j^0(k|k)-\hat{x}_j(k|k)\right]^T\right]\mu_{i|j}(k) \\ &\qquad j=1,\ldots,s \end{aligned} \tag{3.35}$$

The next step involves the filtering calculation for each healthy and faulty mode, respectively. The predicted state for each model and its associated covariance matrix is given in Eq. (3.36).

$$\begin{aligned} \hat{x}_i(k+1|k) &\triangleq E\left[x(k+1)|m_j(k+1),z^k\right] \\ &= F_j(k)\hat{x}_j^0 + G_j(k)u(k) + T_j(k)\bar{\varepsilon}_j(k) \\ P_i(k+1|k) &\triangleq cov\left[\hat{x}_j(k+1|k)|m_j(k+1),z^k\right] \\ &= F_j(k)P_j^0(k|k)F_j(k)^T + T_j(k)Q_j(k)T_j(k)^T \qquad j=1,\ldots,s \end{aligned} \tag{3.36}$$

The measurement residual covariance is computed using Eq. (3.37).

$$\begin{aligned} v_j(k+1) &\triangleq z(k+1) - E\left[z(k+1)|m_j(k+1), z^k\right] \\ &= z(k+1) - H_j(k+1)\hat{x}_i(k+1|k) - D_j(k+1)u(k) - \bar{\eta}(k+1) \\ &\qquad j = 1, \ldots, s \end{aligned} \tag{3.37}$$

and the residual covariance and filter gain are obtained as Eq. (3.38).

$$\begin{aligned} S_j(k+1) &\triangleq cov\left[v_j(k+1|k)|m_j(k+1), z^k\right] \\ &= H_j(k+1)P_j(k+1|k)H_j(k+1)^T + R_j(k+1) \\ K_j(k+1) &= P_j(k+1|k)H_j(k+1)^T S_j(k+1)^{-}1 \qquad j = 1, \ldots, s \end{aligned} \tag{3.38}$$

Using the above residuals, the updated state for each mode (healthy or faulty) is given in Eq. (3.39).

$$\begin{aligned} \hat{x}_j(k+1|k+1) &\triangleq E\left[x(k+1)|m_j(k+1), z^{k+1}\right] \\ &= \hat{x}_i(k+1|k) + K_j(k+1)v_j(k+1) \\ P_j(k+1|k+1) &\triangleq cov\left[\hat{x}_j(k+1|k+1)|m_j(k+1), z^k\right] \\ &= P_j(k+1|k) - K_j(k+1)S_j(k+1)K_j(k+1)^T \qquad j = 1, \ldots, s \end{aligned} \tag{3.39}$$

The equations (3.36) to (3.39) form the model conditional filtering for each mode. The third step involves the mode probability update for FDD decision making. The likelihood function at each step is given in Eq. (3.40).

$$\begin{aligned} L_j(k+1|k+1) &\triangleq N\left[v_j(k+1), 0, S_j(k+1)\right] \\ &= \frac{1}{\sqrt{|(2\pi)S_j(k+1)|}} e^{\left[-\frac{1}{2}v_j(k+1)^T S_j(k+1)^{-1} v_j(k+1)\right]} \qquad j = 1, \ldots, s \end{aligned} \tag{3.40}$$

The mode probability is obtained as Eq. (3.41).

$$\mu_j(k+1) \triangleq P\{m_j(k+1)|z^k\} = \frac{\mu_j(k+1|k)L_j(k+1)}{\sum\limits_{i=1}^{N}\mu_i(k+1|k)L_i(k+1)} \qquad j=1,\ldots,s \tag{3.41}$$

The fault decision logic is obtained using the Eq. (3.42).

$$\frac{\mu_p(k+1)}{max(\mu_i(k+1))} \begin{cases} \geq \mu_T' = H_j : fault \;\; j \;\; occurred \\ \\ < \mu_T' = H_1 : no \;\; fault \end{cases} \qquad i=1,\ldots,p-1,p+1,\ldots,s \tag{3.42}$$

where $\mu_p(k+1)$ is:

$$\mu_p(k+1) = max(\mu_i(k+1)) \qquad i=1,\ldots,s \tag{3.43}$$

The following equations provide the combination of estimates of overall estimates and its covariance.

$$\hat{x}(k+1|k+1) \triangleq E\left[x(k+1)|z^{k+1}\right] = \sum_{j=1}^{s}\hat{x}_j(k+1|k+1)\mu_j(k+1)$$

$$\begin{aligned} P(k+1|k+1) \triangleq E\,[&[x(k+1)-\hat{x}(k+1|k+1)] \\ &[x(k+1)-\hat{x}(k+1|k+1)]^T|z^{k+1}] \\ =& \sum_{j=1}^{s}\left[P_j(k+1|k+1)+\left[\hat{x}(k+1|k+1)-\hat{x}_j(k+1|k+1)\right]\right. \\ &\left.\left[\hat{x}(k+1|k+1)-\hat{x}_j(k+1|k+1)\right]^T\right]\mu_j(k+1) \end{aligned} \tag{3.44}$$

Equations can be divided into two sets; the *time update* equations and the *measurement update* equations. The time update equations which can be distinguished by

$''(k+1|k)''$ subscription are responsible to obtain *a priori* estimates by moving the state and error covariances one step ahead in the time domain. While the measurement update equations, shown by $''(k+1|k+1)''$ subscription are responsible to obtain *a posteriori* estimates through feed-backing measurements into the *a priori* estimates.

This summarizes the complete cycle of the IMM-based FDD scheme using Kalman filters as its mode matched filters. Equation (3.42) not only provides fault detection but also provides isolation, magnitude and fault occurrence time.

3.4 Chapter Summary

In this chapter three different FDD methods named Two-Stage Kalman Filter (TSKF), Adaptive Two-Stage Kalman Filter (ATSKF) and Interacting Multiple Model (IMM) have been presented. They are mainly aimed for application to Unmanned Aerial Vehicles (UAV) and Space missions to detect, isolate and identify possible actuator faults. Next chapter will present some control algorithms for application to unmanned systems.

Chapter 4

Control Algorithms

In this chapter the proposed control algorithms for unmanned systems are presented. First trajectory tracking control of wheeled mobile robots is discussed. A new controller called "Lyapunov-based Guidance Control" (LGC) method is presented. For validation and comparison purposes, a Linear State Tracking Controller (LSTC), a Nonlinear State Tracking controller (NSTC) and a Model Predictive Controller (MPC) are used to validate the proposed control methodology (Lyapunov-based Guidance). For the convenience of reading, a brief description of these algorithms is presented in this chapter. Then the proposed stable formation controller for time-varying configurations of multiple mobile robots is presented. And finally designed Fault Tolerant fuzzy Gain-Scheduled PID (FGS-PID) controller is discussed.

4.1 Trajectory Tracking Control of Wheeled Mobile Robots

4.1.1 Tracking Control Using Lyapunov-Based Guidance Method

In this subsection, design of the proposed controller for trajectory tracking of Wheeled Mobile Robot (WMR) is described in details. As observed in Fig. 4.1, the actual and desired translational/angular positions of the robot have been shown by $(X(t), Y(t), \theta(t))$ and

$(X_r(t), Y_r(t), \theta_r(t))$ respectively. In this figure Virtual Reference Robot shows the desired location and orientation of the Real Robot on the Reference Trajectory.

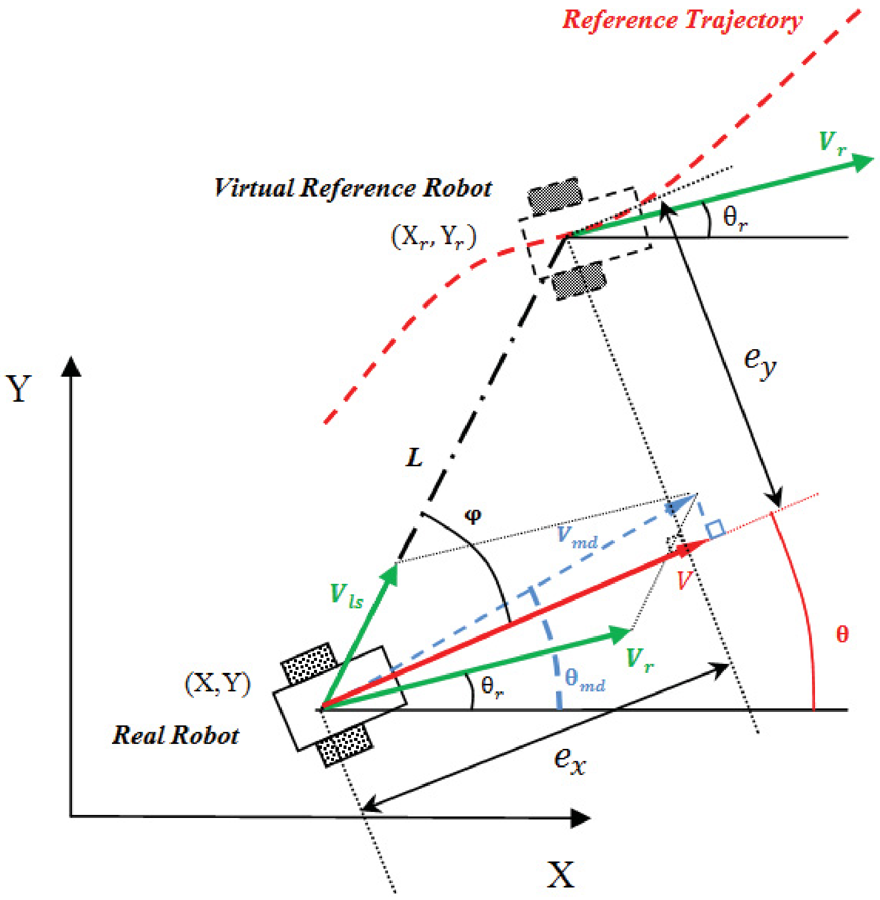


Figure 4.1: The schematic of the robot tracking the reference trajectory. Virtual reference robot shows the desired location and orientation of the real robot on the reference trajectory.

The distance from the real position to the reference position is denoted by L, which is called as "*Line of Sight*, as given in (4.1):

$$L = \sqrt{(X_r - X)^2 + (Y_r - Y)^2} = \sqrt{e_x^2 + e_y^2} \tag{4.1}$$

The angle between the line of sight and x axis is called as "*Angle of Sight*, as given in (4.2):

$$\begin{aligned}\varphi &= atan2((Y_r - Y),(X_r - X)) - \theta \\ &= atan2(e_y, e_x)\end{aligned} \tag{4.2}$$

The error in polar coordination system is defined as in (4.3):

$$\mathbf{q_e} = \begin{bmatrix} L \\ \varphi \\ e_\theta \end{bmatrix} = \begin{bmatrix} \sqrt{e_x^2 + e_y^2} \\ atan2(e_y, e_x) \\ \theta_r - \theta \end{bmatrix} \tag{4.3}$$

Assuming the perfect velocity tracking, robot's kinematic equation can be rewritten as the form in (4.4):

$$\dot{\mathbf{q}}_\mathbf{e} = \begin{bmatrix} \dot{L} \\ \dot{\varphi} \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} cos(\varphi)(v_r cos(e_\theta) - v) + L sin(\varphi) sin(e_\theta) \\ v sin(\varphi) + L\omega - v_r sin(\varphi - e_\theta) \\ \omega_r - \omega \end{bmatrix} \tag{4.4}$$

The control objective is to make the error vector approach to zero, or in other words the control objective is to use two control inputs (v, ω) for making the mobile robot to track the reference/desired trajectory.

Based on what has been stated, when L approaches zero and θ approaches θ_r, then the total error of the system will be zero and the desired trajectory would be tracked.

A common and simple solution for trajectory tracking problem could be pursuit guidance [2], in which the vehicle is always driven to its current desired position in reference trajectory. In other words, during the tracking procedure, desired heading angle of the robot is always set to be its angle of sight (φ). Although pursuit guidance algorithm is quite straightforward and easy to use, it is not satisfactory in terms of tracking performance in applications with high manoeuvrability.

Inspired by our previous work [3], *to increase tracking performance, the control inputs should not necessarily take the vehicle as close as possible to its current desired position, but they should make the robot approach to the desired trajectory*. In other words, the controlled system should have a sense of possible future trajectory and devise a timely and appropriate policy to catch the desired trajectory as fast as possible.

To this end, a good strategy is to continuously reduce the distance L which is more in line with human's natural behaviour and hence a more intelligent method to be pursued here. To make the distance L approaches zero, one can imagine of two different components for linear velocity. To keep the robot in the vicinity of desired trajectory, linear velocity of reference trajectory (v_r) should be considered as one of the components (See Fig. 4.1). To decrease distance L, another component should be toward the desired trajectory and in the direction of line of sight. This component can be simply achieved through a simple proportional relation to distance L as follows (See Fig. 4.1):

$$v_{ls} = K_v L \tag{4.5}$$

where K_v is a strictly positive constant. Now, the vector summation of these two components could be counted as a proper linear velocity of the robot (4.6).

$$\begin{aligned} \vec{v}_{md} &= \vec{v}_r + \vec{v}_{ls} \\ v_{md} &= \sqrt{v_r^2 + v_{ls}^2 + 2v_r v_{ls} cos(\varphi - e_\theta)} \end{aligned} \tag{4.6}$$

where v_{md} is the "*modified desired*" linear velocity of the robot. Since the robot is of non-holonomic type, due to geometrical constraint, the computed linear velocity v_{md} cannot be rendered, unless current heading angle of the robot (θ) changes to the direction of modified desired velocity, which is called "*modified desired*" heading angle (θ_{md}).

Based on the strategy discussed above, "*modified desired*" heading angle can be achieved through (4.7).

$$\theta_{md} = atan2\left(\frac{v_{ls}sin(\varphi - e_\theta)}{v_r - v_{ls}cos(\varphi - e_\theta)}\right) + \theta_r \tag{4.7}$$

It is worthy to note that, based on (4.7), when L approaches zero, θ_{md} approaches θ_r, otherwise desired tracking will not be achieved. Considering θ_{md} as a new heading objective one can define a new heading error as (4.8).

$$\acute{e}_\theta = \theta_{md} - \theta \tag{4.8}$$

The effective linear velocity is calculated as the projection of the two discussed components into the robots heading direction (4.9).

$$v = v_{ls}cos(\varphi) + v_r cos(e_\theta) \tag{4.9}$$

Replacing v_{ls} from (4.5) into (4.10) yields:

$$v = K_v L cos(\varphi) + v_r cos(e_\theta) \tag{4.10}$$

Now the remaining part of the control problem is to choose angular velocity (ω) to stabilize the closed-loop controlled system. Angular velocity is proposed as (4.11).

$$\omega = \dot{\theta}_{md} + v_{md}\left(K_\omega(v_{ls}sin(\varphi) + v_r sin(e_\theta)) + Lsin(\varphi)\right) \tag{4.11}$$

where K_ω is a strictly positive constant. To calculate input angular velocity, (4.7) is derived as in (4.12):

$$\begin{aligned}\dot{\theta}_{md} = &[(\dot{v}_{ls}sin(\varphi - e_\theta) + v_{ls}(\dot{\varphi} - \dot{e}_\theta)cos(\varphi - e_\theta))\\ &(v_r + v_{ls}cos(\varphi - e_\theta))]/v_{md}^2 - [(v_{ls}sin(\varphi - e_\theta))\\ &(\dot{v}_r + \dot{v}_{ls}cos(\varphi - e_\theta) - v_{ls}(\dot{\varphi} - \dot{e}_\theta)sin(\varphi - e_\theta))]/v_{md}^2 + \omega_r\end{aligned} \tag{4.12}$$

which yields to (4.13):

$$\begin{aligned}\dot{\theta}_{md} = &[(v_r\dot{v}_{ls} - \dot{v}_r v_{ls})sin(\varphi - e_\theta) + v_{ls}^2(\dot{\varphi} - \dot{e}_\theta)\\ &+ v_r v_{ls}(\dot{\varphi} - \dot{e}_\theta)cos(\varphi - e_\theta)]/v_{md}^2 + \omega_r\end{aligned} \tag{4.13}$$

Eq. (4.13) can be rewritten in the form of (4.14).

$$\dot{\theta}_{md} = \beta_1\omega + \beta_2 \tag{4.14}$$

where β_1 and β_2 are:

$$\begin{aligned}\beta_1 &= (v_r v_{ls}cos(\varphi - e_\theta) + v_{ls}^2)/v_{md}^2\\ \beta_2 &= [(v_r\dot{v}_{ls} - \dot{v}_r v_{ls})sin(\varphi - e_\theta) + v_{ls}^2(\dot{\varphi} - \omega_r)\\ &\quad + v_r v_{ls}(\dot{\varphi} - \omega_r)cos(\varphi - e_\theta)]/v_{md}^2\end{aligned} \tag{4.15}$$

where $\dot{v}_{ls} = K_v(v_r sin(\varphi)sin(e_\theta) - K_v L cos^2(\varphi))$. Substituting (4.14) into (4.11) and solving for ω, yields (4.16).

$$\omega = [\beta_2 + v_{md}\left(K_\omega(v_{ls}sin(\varphi) + v_r sin(e_\theta)) + Lsin(\varphi)\right)]/(1 - \beta_1) \tag{4.16}$$

Theorem 1. Using (4.10) and (4.16) as control inputs, the closed-loop control system is stable.

Proof: Taking Lyapunov candidate function as (4.17),

$$
\begin{aligned}
V &= \tfrac{1}{2}L^2 + 1 - cos\left(atan2\left(\frac{K_v Lsin(\varphi - e_\theta)}{v_r + K_v Lcos(\varphi - e_\theta)}\right) + e_\theta\right) \\
&= \tfrac{1}{2}L^2 + 1 - cos(\theta_{md} - \theta)
\end{aligned}
\tag{4.17}
$$

where V is a positive definite function ($V \geq 0$). $V = 0$ if and only if $\mathbf{q_e} = 0$. If $\mathbf{q_e} > 0$, then $V > 0$ and vice versa.

Deriving Lyapunov candidate function:

$$
\begin{aligned}
\dot{V} &= \dot{L}L + (\dot{\theta}_{md} - \omega)sin(\theta_{md} - \theta) \\
\\
&= (e_y\omega - v + v_r cos(e_\theta))e_x + (-e_x\omega + v_r sin(e_\theta))e_y \\
&\quad + (\dot{\theta}_{md} - \omega)\ sin\left(atan2\left(\frac{v_{ls}sin(\varphi - e_\theta)}{v_r + v_{ls}cos(\varphi - e_\theta)}\right) + e_\theta\right) \\
\\
&= Lcos(\varphi)(v_r cos(e_\theta) - v) + Lv_r sin(\varphi)sin(e_\theta) \\
&\quad + [((\beta_1 - 1)\omega + \beta_2)(v_{ls}sin(\varphi - e_\theta)cos(e_\theta) \\
&\quad + (v_r + v_{ls}cos(\varphi - e_\theta))sin(e_\theta))]/v_{md} \\
\\
&= Lcos(\varphi)(v_r cos(e_\theta) - v) + Lv_r sin(\varphi)sin(e_\theta) \\
&\quad + [((\beta_1 - 1)\omega + \beta_2)(v_r sin(e_\theta) + v_{ls}sin(\varphi))]/v_{md}
\end{aligned}
\tag{4.18}
$$

Substituting v_{ls}, v, w from (4.5), (4.10), and (4.16) into (4.18):

$$
\begin{aligned}
\dot{V} &= Lcos(\varphi)(v_r cos(e_\theta) - K_v Lcos(\varphi) - v_r cos(e_\theta)) \\
&\quad + Lv_r sin(\varphi)sin(e_\theta) \\
&\quad + (k_\omega(v_{ls}sin(\varphi) + v_r sin(e_\theta)) + Lsin(\varphi))(k_v Lsin(\varphi) + v_r sin(e_\theta)) \\
\\
&= -K_v L^2 - K_\omega(K_v Lsin(\varphi) + v_r sin(e_\theta))^2 \leqslant 0
\end{aligned}
\tag{4.19}
$$

where in (4.19), $\dot{V} = 0$ if and only if $\dot{\mathbf{q}}_{\mathbf{e}} = 0$. Based on Lyapunov theory [97], V is a Lyapunov function and the closed-loop control system is asymptotically stable.

4.1.2 Model Predictive Controller

Benefited by its unique features such as feasibility and stability, MPC has been widely and successfully applied to systems with physical constraints. The objective of the MPC is to drive the predicted robot trajectory as close as possible to the future reference trajectory, i.e., to track the reference trajectory [16].

The robot's kinematic (2.10) can be rewritten as (4.20):

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} cos(e_\theta) & 0 \\ sin(e_\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{4.20}$$

In (4.20) controlled input signals (v and ω) can be written as a combination of a feedforward and a feedback commands as shown in (4.21).

$$\begin{cases} v = v_f + v_b \\ \omega = \omega_f + \omega_b \end{cases} \tag{4.21}$$

where $v_f = v_r cos(e_\theta)$ and $\omega_f = \omega_r$. By linearizing (4.20) around a reference trajectory ($e_x = e_y = e_\theta = 0$, $v_b = \omega_b = 0$), the linearized model can be described as (4.22):

$$\dot{\mathbf{p}}_\mathbf{e} = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \mathbf{p}_\mathbf{e} + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v_b \\ \omega_b \end{bmatrix} \tag{4.22}$$

where $\mathbf{p}_\mathbf{e}$ is the error vector ($\mathbf{p}_\mathbf{e} = [e_x e_y e_\theta]^T$). The idea of the moving-horizon control concept is to find the control-variable values that minimize the receding-horizon quadratic cost function (in a certain interval denoted by h) based on the predicted robot-following error (4.23):

$$J(v_b, \omega_b, k) = \sum_{i=1}^{h} \Big([e_r(k+i) - e(k+i|k)]^T Q [e_r(k+i) - e(k+i|k)] \\ + [v_b(k,i) w_b(k,i)] R [v_b(k,i) w_b(k,i)]^T \Big) \tag{4.23}$$

where $e_r(k+i)$ and $e(k+i)$ are the reference robot-following trajectory and the robot-following error, respectively. Q and R are the weighting matrices where $Q \in R^n \times R^n$ and $R \in R^m \times R^m$, with $Q \geq 0$ and $R \geq 0$.

Using discrete linear model of error dynamic, one can calculate the feedback control signal by applying Model Predictive Control Algorithm as (4.24):

$$\begin{bmatrix} v_b \\ \omega_b \end{bmatrix} = (G^T \bar{Q} G + \bar{R})^{-1} G^T \bar{Q} (F_r - F) \mathbf{p_e}(k) \tag{4.24}$$

where $\bar{Q}$, $\bar{R}$, and F_r are design parameters (For detailed description please refer to [16]).

4.1.3 Linear State Tracking Controller

As previously stated in the Introduction, Kanayama *et al.* [10] proposed a controller which is locally stable around reference trajectory. With the system structure described as in Fig. 4.1, Linear State Tracking Control (LSTC) law can be given in (4.25):

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r cos(e_\theta) + K_x e_x \\ \omega_r + sign(v_r) K_y e_y + K_\theta e_\theta \end{bmatrix} \tag{4.25}$$

where K_x,K_y, and K_θ are calculated as (4.26):

$$\begin{cases} K_x = 2\xi \omega_n \\ K_y = (\omega_n^2 - \omega_r^2)/v_r \qquad if \quad v_r > v_{switch} \\ K_\theta = 2\xi \omega_n \end{cases} \tag{4.26}$$

where ω_n and $\xi \in (0,1)$ are strictly positive constants. If the reference velocity goes to zero then the gain K_y may goes to infinity. A possible gain scheduling solution is to use (4.27) in this case.

$$\begin{cases} K_x = 2\xi\sqrt{\omega_r^2 + gv_r^2} \\ K_y = gv_r \qquad\qquad if \quad v_r \leq v_{switch} \\ K_\theta = 2\xi\sqrt{\omega_r^2 + gv_r^2} \end{cases} \tag{4.27}$$

where $g > 0$. Readers are referred to [10] for detailed description.

4.1.4 Nonlinear State Tracking Controller

The main advantage of NSTC over LSTC is that it is globally asymptotically stable. With the same system structure described as in Fig. 4.1, Nonlinear State Tracking Control (NSTC) law can be given in (4.28):

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r cos(e_\theta) + K_x e_x \\ \omega_r + K_y sin(e_\theta) e_y / e_\theta + K_\theta e_\theta \end{bmatrix} \tag{4.28}$$

where K_x,K_y, and K_θ are calculated as (4.29):

$$\begin{cases} K_x = 2\xi\sqrt{\omega_r^2 + gv_r^2} \\ K_y = g \\ K_\theta = 2\xi\sqrt{\omega_r^2 + gv_r^2} \end{cases} \tag{4.29}$$

where $g > 0$. Readers are referred to [14] for detailed description.

4.2 Varying-Configuration Formation Control of Multiple Wheeled Mobile Robots

4.2.1 Modeling of Robot's Motion in the Leader-Follower Approach

Kinematical model of mobile robot is described in 4.1.1. In order to describe the position of each follower relative to the leader, the distance-angle method is adopted. In other words, to express the desired position of each robot member in the group, and hence describe the complete configuration or arrangement of the total robots, it is sufficient that the distance L_{lf} of the follower robot from the leader and the angle φ_{lf} between the heading direction and the line connecting them (known as the line of sight angle) be determined (see Fig. 4.2). Using this geometry, it is possible to describe the various configurations of the robots group.

In Fig. 4.2, v_l denotes the linear velocity of the leader and θ_l represents heading angle of the leader with respect to horizontal direction. It is assumed that, if the leader and follower robots have any given arbitrary configuration relative to each other, then the follower can measure its distance L_{lf} from the leader. Furthermore, the follower can provide the line of sight angle φ_{lf} between itself and the leader.

Separation-bearing is used to describe the desired configuration of each pair of leader-follower in formation. The desired position of the follower robot is shown as *Virtual Follower*. d_{lf} is the desired separation distance between leader and follower and ψ_{lf} is the desired separation angle. To investigate the transition mode both d_{lf} and ψ_{lf} are assumed to be known as a function of time (Fig. 4.3).

The desired linear velocity of follower (or the linear velocity of virtual follower) is the sum of three elements. One is the linear velocity of the leader (v_l), the others are changes in separation distance ($\dot{d}_{lf}$) and the effects of rotational velocity of the leader and changes in bearing angle which are represented as $d(\omega_l + \dot{\psi}_{lf})$. Then the desired linear

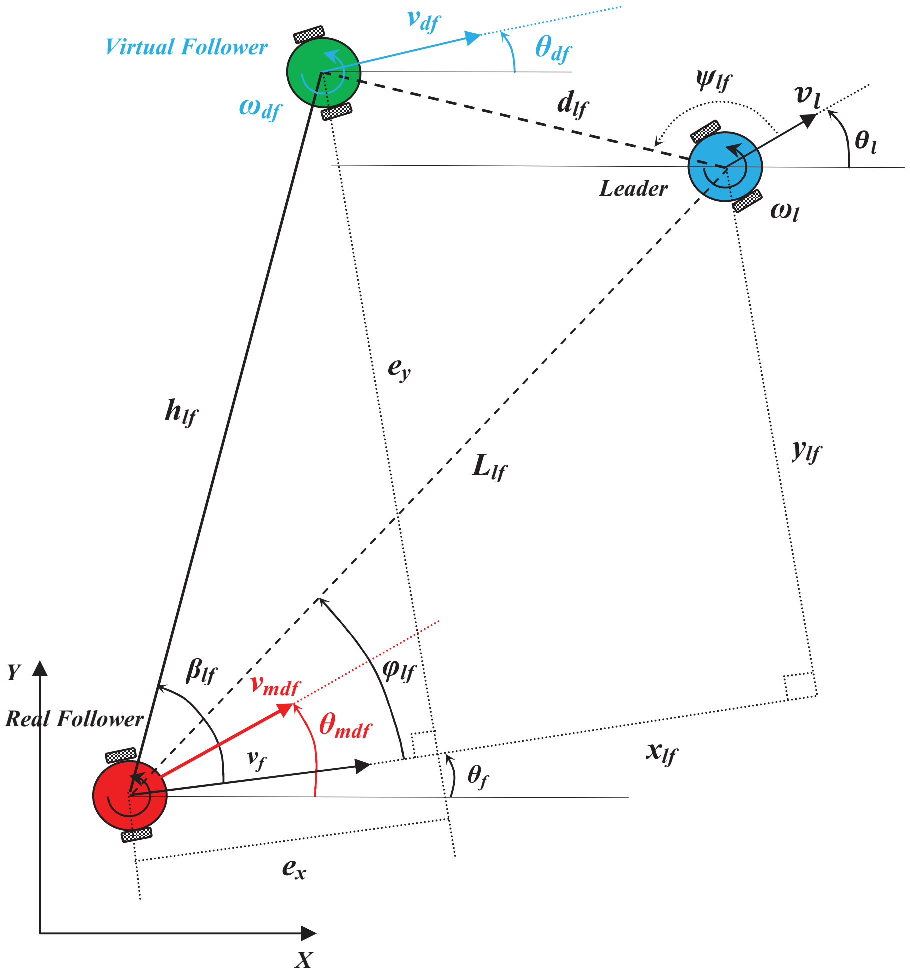


Figure 4.2: Geometry for the relative configuration of the leader and follower robots. The real follower and virtual follower robots indicate the real and desired instantaneous positions of the follower, respectively.

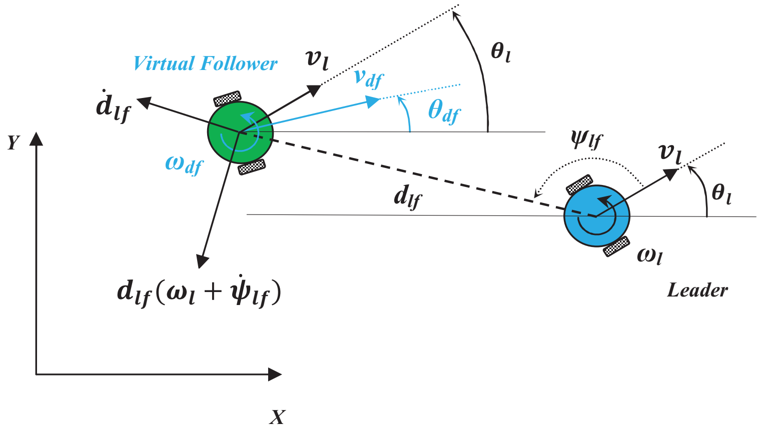


Figure 4.3: Separation-Bearing configuration of leader and virtual follower.

velocity of follower is calculated as Eq. 4.30.

$$v_{df} = \sqrt{(v_l sin(\psi_{lf}) - d(\omega_l + \dot{\psi}_{lf}))^2 + (v_l cos(\psi_{lf}) + \dot{d})^2} \tag{4.30}$$

The desired heading angle of the follower (or the heading angle of virtual follower) is aligned to follower desired linear velocity v_{df}. Based on the represented geometry in Fig. 4.3 it is calculated in Eq. 4.31.

$$\theta_{df} = \theta_l + \psi_{lf} - atan2\left(\frac{v_l sin(\psi_{lf}) - d(\omega_l + \dot{\psi}_{lf})}{v_l cos(\psi_{lf}) + \dot{d}}\right) \tag{4.31}$$

The desired angular velocity of follower robot can also be calculated by deriving 4.31 as shown in Eq. 4.32.

$$\begin{aligned}
\omega_{df} = \dot{\psi} + \omega_l - [&v_l \dot{v}_l sin(\psi_{lf}) cos(\psi_{lf}) + v_l^2 \dot{\psi}_{lf} \\
&- v_l \dot{d}(\omega_l + \dot{\psi}_{lf}) cos(\psi_{lf}) - d v_l(\dot{\omega}_l + \ddot{\psi}_{lf}) cos(\psi_{lf}) \\
&+ \dot{d}\dot{v}_l sin(\psi_{lf}) + v_l \dot{d}\dot{\psi}_{lf} cos(\psi_{lf}) - \dot{d}^2(\omega_l + \dot{\psi}_{lf}) \\
&- d\dot{d}(\dot{\omega}_l + \ddot{\psi}_{lf}) - v_l \dot{v}_l sin(\psi_{lf}) cos(\psi_{lf}) \\
&+ v_l^2 \dot{\psi}_{lf} sin^2(\psi_{lf}) \\
&- v_l \ddot{d} sin(\psi_{lf}) + d\dot{v}_l(\omega_l + \dot{\psi}_{lf}) cos(\psi_{lf}) \\
&- d v_l(\omega_l + \dot{\psi}_{lf})\dot{\psi}_{lf} sin(\psi_{lf}) + d\ddot{d}(\omega_l + \dot{\psi}_{lf})]/v_{df}^2
\end{aligned} \tag{4.32}$$

4.2.2 Formation Controller

In this subsection, design of the proposed formation controller is described in details. As observed in Fig. 4.2, the actual and desired translational/angular positions of the follower robot have been shown by $(X_f(t), Y_f(t), \theta_f(t))$ and $(X_{df}(t), Y_{df}(t), \theta_{df}(t))$ respectively.

The distance from the position of the real follower to the position of virtual follower is denoted by h_{lf}, which is called as "*Virtual Line of Sight*", as given in (4.33):

$$h_{lf} = \sqrt{(X_{df} - X_f)^2 + (Y_{df} - Y_f)^2} = \sqrt{e_x^2 + e_y^2} \tag{4.33}$$

The angle between the virtual line of sight and x-axis is called as "*Virtual Angle of Sight*, as given in (4.34):

$$\begin{aligned} \beta_{lf} &= atan2((Y_{df} - Y_f), (X_{df} - X_f)) - \theta_f \\ &= atan2(e_y, e_x) \end{aligned} \tag{4.34}$$

The error in polar coordination system is defined as in (4.35):

$$\mathbf{q_e} = \begin{bmatrix} h_{lf} \\ \beta_{lf} \\ e_\theta \end{bmatrix} = \begin{bmatrix} \sqrt{e_x^2 + e_y^2} \\ atan2(e_y, e_x) \\ \theta_{df} - \theta_f \end{bmatrix} \tag{4.35}$$

Assuming the perfect velocity tracking, robot's kinematic equation can be rewritten as the form in (4.36):

$$\dot{\mathbf{q}}_\mathbf{e} = \begin{bmatrix} \dot{h_{lf}} \\ \dot{\beta_{lf}} \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} cos(\beta_{lf})(v_{df}cos(e_\theta) - v_f) + h_{lf}sin(\beta_{lf})sin(e_\theta) \\ v_f sin(\beta_{lf}) + h_{lf}\omega_f - v_{df}sin(\beta_{lf} - e_\theta) \\ \omega_{lf} - \omega_f \end{bmatrix} \tag{4.36}$$

The formation control objective is to make the error vector approach to zero, or in other words to use two control inputs (v_f, ω_f) for making the real follower to track the virtual one.

Based on what has been stated, when h_{lf} approaches zero and θ_f approaches θ_{lf}, then the total error of the system will be zero and formation is shaped.

To this end, a good strategy is to continuously reduce the distance h_{lf} which is more in line with human's natural behaviour and hence a more intelligent method to be pursued

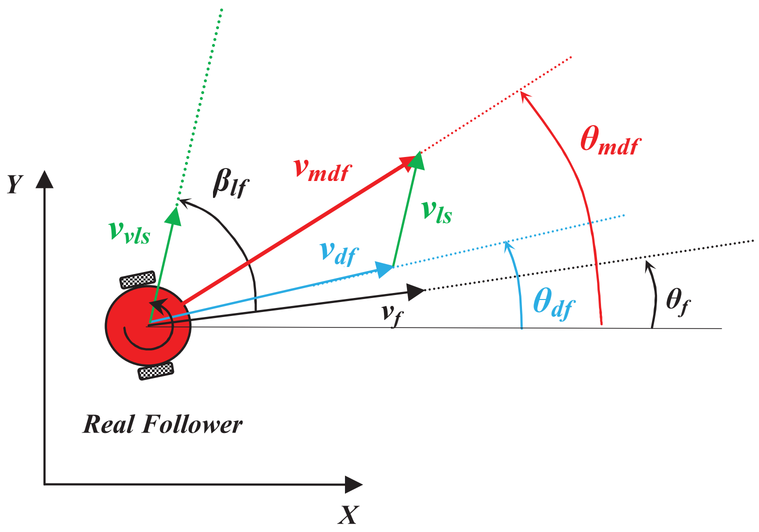


Figure 4.4: The schematic of the real follower and its desired velocity components.

here. To make the distance h_{lf} approaches zero, one can imagine of two different components for linear velocity. To keep the follower robot in the vicinity of its virtual one, linear velocity of virtual follower (v_{df}) should be considered as one of the components (See Fig. 4.4). To decrease distance h_{lf}, another component should be toward the virtual follower and in the direction of virtual line of sight. This component can be simply achieved through a simple proportional relation to distance h_{lf} as follows (See Fig. 4.4):

$$v_{vls} = K_v h_{lf} \tag{4.37}$$

where K_v is a strictly positive constant. Now, the vector summation of these two components could be counted as a proper linear velocity of the real follower (4.38).

$$\begin{gathered}\vec{v}_{mdf} = \vec{v}_{df} + \vec{v}_{vls} \\ v_{mdf} = \sqrt{v_{df}^2 + v_{vls}^2 + 2v_{df}v_{vls}cos(\beta_{lf} - e_\theta)}\end{gathered} \tag{4.38}$$

where v_{mdf} is the "*modified desired*" linear velocity of the real follower. Since the robots are of nonholonomic type, due to geometrical constraint, the computed linear velocity (v_{mdf}) cannot be rendered, unless current heading angle of the follower (θ) changes to the direction of modified desired velocity, which is called "*follower's modified desired heading angle*" (θ_{mdf}).

Based on the strategy discussed above, "*modified desired*" heading angle can be achieved through (4.39).

$$\theta_{mdf} = atan2\left(\frac{v_{vls}sin(\beta_{lf} - e_\theta)}{v_{df} - v_{vls}cos(\beta_{lf} - e_\theta)}\right) + \theta_{df} \tag{4.39}$$

It is worthy to note that, based on (4.7), when h_{lf} approaches zero, θ_{mdf} approaches θ_{df}, otherwise desired formation will not be achieved. Considering θ_{mdf} as a new heading objective one can define a new heading error as (4.40).

$$\acute{e}_\theta = \theta_{mdf} - \theta_f \tag{4.40}$$

The effective linear velocity is calculated as the projection of the two discussed components into the robot's heading direction (4.41).

$$v_f = v_{vls}cos(\beta_{lf}) + v_{df}cos(e_\theta) \tag{4.41}$$

Replacing v_{vls} from (4.5) into (4.42) yields:

$$v_f = K_v h_{lf} cos(\beta_{lf}) + v_{df}cos(e_\theta) \tag{4.42}$$

Now the remaining part of the control problem is to choose angular velocity (ω_f) to stabilize the closed-loop controlled system. Angular velocity is proposed as (4.43):

$$\omega_f = \dot{\theta}_{mdf} + v_{mdf}\left(K_\omega(v_{vls}sin(\beta_{lf}) + v_{df}sin(e_\theta)) + h_{lf}sin(\beta_{lf})\right) \tag{4.43}$$

where K_ω is a strictly positive constant. To calculate input angular velocity, (4.7) is derived as in (4.44):

$$\begin{aligned}\dot{\theta}_{mdf} = [\Big(&\dot{v}_{vls}sin(\beta_{lf}-e_\theta)+v_{vls}(\dot{\beta}_{lf}-\dot{e}_\theta)cos(\beta_{lf}-e_\theta)\Big)\\ &\Big(v_{df}+v_{vls}cos(\beta_{lf}-e_\theta)\Big)]/v_{mdf}^2-[\Big(v_{vls}sin(\beta_{lf}-e_\theta)\Big)\\ &\Big(\dot{v}_{df}+\dot{v}_{vls}cos(\beta_{lf}-e_\theta)-v_{vls}(\dot{\beta}_{lf}-\dot{e}_\theta)sin(\beta_{lf}-e_\theta)\Big)]/v_{mdf}^2+\omega_{df}\end{aligned}\tag{4.44}$$

which yields to (4.45):

$$\begin{aligned}\dot{\theta}_{mdf} = [&(v_{df}\dot{v}_{vls}-\dot{v}_{df}v_{vls})sin(\beta_{lf}-e_\theta)+v_{vls}^2(\dot{\beta}_{lf}-\dot{e}_\theta)\\ &+v_{df}v_{vls}(\dot{\beta}_{lf}-\dot{e}_\theta)cos(\beta_{lf}-e_\theta)]/v_{mdf}^2+\omega_{df}\end{aligned}\tag{4.45}$$

Eq. (4.13) can be rewritten in the form of (4.46).

$$\dot{\theta}_{mdf} = \alpha_1\omega+\alpha_2 \tag{4.46}$$

where α_1 and α_2 are:

$$\begin{aligned}\alpha_1 &= (v_{df}v_{vls}cos(\beta_{lf}-e_\theta)+v_{vls}^2)/v_{mdf}^2\\ \\ \alpha_2 &= [(v_{df}\dot{v}_{vls}-\dot{v}_{df}v_{vls})sin(\beta_{lf}-e_\theta)+v_{vls}^2(\dot{\beta}_{lf}-\omega_{df})\\ &\quad+v_{df}v_{vls}(\dot{\beta}_{lf}-\omega_{df})cos(\beta_{lf}-e_\theta)]/v_{mdf}^2\end{aligned}\tag{4.47}$$

where $\dot{v}_{vls} = K_v(v_{df}sin(\beta_{lf})sin(e_\theta)-K_vh_{lf}cos^2(\beta_{lf}))$. Substituting (4.14) into (4.11) and solving for ω_f, yields (4.48):

$$\begin{aligned}\omega_f &= \alpha_2/(1-\alpha_1)\\ &\quad+v_{mdf}\left[K_\omega(v_{vls}sin(\beta_{lf})+v_{df}sin(e_\theta))\right]/(1-\alpha_1)\\ &\quad+h_{lf}sin(\beta_{lf})/(1-\alpha_1)\end{aligned}\tag{4.48}$$

Theorem 1. Using (4.10) and (4.16) as control inputs, the closed loop control system is stable.

Proof: Taking Lyapunov candidate function as (4.49):

$$
\begin{aligned}
V &= \tfrac{1}{2}h_{lf}^2 + 1 - cos\left(atan2\left(\frac{K_v h_{lf} sin(\beta_{lf} - e_\theta)}{v_{md} + K_v h_{lf} cos(\beta_{lf} - e_\theta)}\right) + e_\theta\right) \\
&= \tfrac{1}{2}h_{lf}^2 + 1 - cos(\theta_{mdf} - \theta)
\end{aligned}
\tag{4.49}
$$

where V is a positive definite function ($V \geq 0$). $V = 0$ if and only if $\mathbf{q_e} = 0$. If $\mathbf{q_e} > 0$, then $V > 0$ and vice versa.

Deriving Lyapunov candidate function:

$$
\begin{aligned}
\dot{V} &= \dot{h_{lf}} h_{lf} + (\dot{\theta}_{mdf} - \omega_f) sin(\theta_{mdf} - \theta_f) \\
\\
&= (e_y \omega_f - v_f + v_{df} cos(e_\theta)) e_x \\
&\quad + (-e_x \omega_f + v_{df} sin(e_\theta)) e_y + (\dot{\theta}_{mdf} - \omega_f) \\
&\quad sin\left(atan2\left(\frac{v_{vls} sin(\beta_{lf} - e_\theta)}{v_{df} + v_{vls} cos(\beta_{lf} - e_\theta)}\right) + e_\theta\right) \\
\\
&= h_{lf} cos(\beta_{lf})(v_{df} cos(e_\theta) - v_f) \\
&\quad + h_{lf} v_{df} sin(\beta_{lf}) sin(e_\theta) \\
&\quad + [((\alpha_1 - 1)\omega_f + \alpha_2)(v_{vls} sin(\beta_{lf} - e_\theta) cos(e_\theta) \\
&\quad + (v_{df} + v_{vls} cos(\beta_{lf} - e_\theta)) sin(e_\theta))] / v_{mdf} \\
\\
&= h_{lf} cos(\beta_{lf})(v_{df} cos(e_\theta) - v_f) \\
&\quad + h_{lf} v_{df} sin(\beta_{lf}) sin(e_\theta) \\
&\quad + [((\alpha_1 - 1)\omega_f + \alpha_2)(v_{df} sin(e_\theta) + v_{vls} sin(\beta_{lf}))] / v_{mdf}
\end{aligned}
\tag{4.50}
$$

Substituting v_{vls}, v_f, ω_f from (4.5), (4.10), and (4.16) into (4.18):

$$\begin{aligned}\dot{V} &= h_{lf}cos(\beta_{lf})\\ &\quad (v_{df}cos(e_\theta) - K_v h_{lf}cos(\beta_{lf}) - v_{lf}cos(e_\theta))\\ &\quad + h_{lf}v_{df}sin(\beta_{lf})sin(e_\theta)\\ &\quad + (k_\omega(v_{vls}sin(\beta_{lf}) + v_{df}sin(e_\theta)) + h_{lf}sin(\beta_{lf}))\\ &\quad (k_v h_{lf}sin(\beta_{lf}) + v_{df}sin(e_\theta))\end{aligned} \tag{4.51}$$

$$= -K_v h_{lf}^2 - K_\omega(K_v h_{lf}sin(\beta_{lf}) + v_{df}sin(e_\theta))^2 \leqslant 0$$

where in (4.19), $\dot{V} = 0$ if and only if $\dot{\mathbf{q}}_\mathbf{e} = 0$. Based on Lyapunov theory [97], V is a Lyapunov function and the closed-loop control system is asymptotically stable.

4.3 Fault Tolerant Fuzzy Gain-Scheduled PID Control for Quadrotor UAV

Conventional PID controllers are frequently and widely used in vast number of industrial applications. They are simple and easy to use due to the fact that they do not need any mathematical model of the controlled process or complicated theories. But one of the main drawbacks of these controllers is that there is no certain way for choosing the control parameters which guarantees good performance of the system.

The transfer function of a conventional PID controller is:

$$G(s) = K_p + \frac{K_i}{s} + K_d s \tag{4.52}$$

where K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively. Although PID controllers are robust against structural changes and uncertainties in the system parameters, their performance may be affected by such changes or may even lead to system instability. Therefore in real world applications these gains need to be fine-tuned to keep the required performance. To overcome this shortcoming, Fuzzy Logic Controller (FLC) is used to tune PID gains online where the tracking error and the change of the tracking error are used to determine control parameters. Fig. 4.5 shows a schematic of Fuzzy Gain Scheduling PID controller.

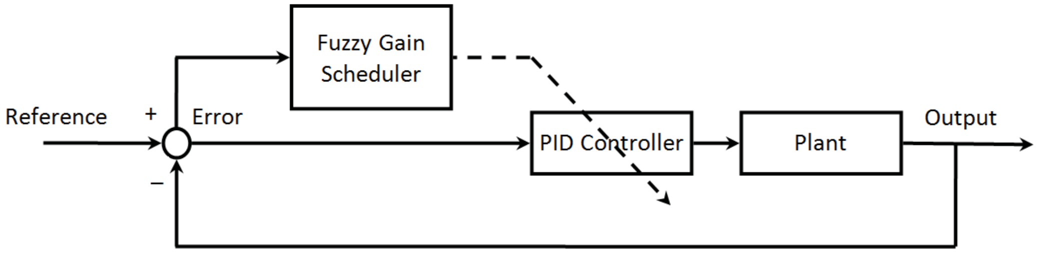


Figure 4.5: Fuzzy gain scheduling scheme for PID controller.

Controller gains can be calculated through a simple linear transformation:

$$K_p = \left(K_{p,\max} - K_{p,\min}\right) K'_p + K_{p,\min} \tag{4.53}$$

$$K_i = \left(K_{i,\max} - K_{i,\min}\right) K'_i + K_{i,\min} \tag{4.54}$$

$$K_d = \left(K_{d,\max} - K_{d,\min}\right) K'_d + K_{d,\min} \tag{4.55}$$

with $\left[K_{p,\min}, K_{p,\max}\right]$, $\left[K_{i,\min}, K_{i,\max}\right]$ and $\left[K_{d,\min}, K_{d,\max}\right]$ are predefined ranges of K_p, K_i, and K_d respectively. A set of linguistic rules in the form of (4.56) is used in the FLC structure to determine K'_p, K'_i and K'_d:

$$\text{If } e(k) \text{ is } A_i \text{ and } \Delta e(k) \text{ is } B_i \text{ then } K'_p \text{ is } C_i,\ K'_i \text{ is } D_i, \text{ and } K'_d \text{ is } E_i \tag{4.56}$$

where A_i, B_i, C_i, D_i, and E_i are fuzzy sets corresponding to $e(k)$, $\Delta e(k)$, K'_p, K'_i, and K'_d respectively. Three sets of 49 rules are used to determine controller gains. Tables 4.1-4.3 show the linguist rules used in the FLC. In these tables, N, P, ZO, S, M, B represent negative, positive, approximately zero, small, medium, and big respectively. For example NB means Negative Big, and so on.

Table 4.1: Fuzzy tuning rules for K_p'

		$\Delta e(k)$						
		NB	**NM**	**NS**	**ZO**	**PS**	**PM**	**PB**
$e(k)$	**NB**	B	B	B	B	B	B	B
	NM	S	B	B	B	B	B	S
	NS	S	S	B	B	B	S	S
	ZO	S	S	S	B	S	S	S
	PS	S	S	B	B	B	S	S
	PM	S	B	B	B	B	B	S
	PB	B	B	B	B	B	B	B

Table 4.2: Fuzzy tuning rules for K_i'

		$\Delta e(k)$						
		NB	**NM**	**NS**	**ZO**	**PS**	**PM**	**PB**
$e(k)$	**NB**	S	S	S	S	S	S	S
	NM	B	B	S	S	S	B	B
	NS	B	B	B	S	B	B	B
	ZO	B	B	B	B	B	B	B
	PS	B	B	B	S	B	B	B
	PM	B	B	S	S	S	B	B
	PB	S	S	S	S	S	S	S

The membership functions for input variables are defined with triangular and trapezoidal shapes and those for output variables are singleton (Figures 4.6 and 4.7). All the fuzzy sets for input and output values are normalized for convenience.

The generated surfaces for the FLC are shown in Figures 4.8-4.10.

Table 4.3: Fuzzy tuning rules for K_d'

		$\Delta e(k)$						
		NB	**NM**	**NS**	**ZO**	**PS**	**PM**	**PB**
	NB	B	B	B	B	B	B	B
	NM	M	M	B	B	B	M	M
	NS	S	M	M	B	M	M	S
$e(k)$	**ZO**	ZO	S	M	B	M	S	ZO
	PS	S	M	M	B	M	M	S
	PM	M	M	B	B	B	M	M
	PB	B	B	B	B	B	B	B

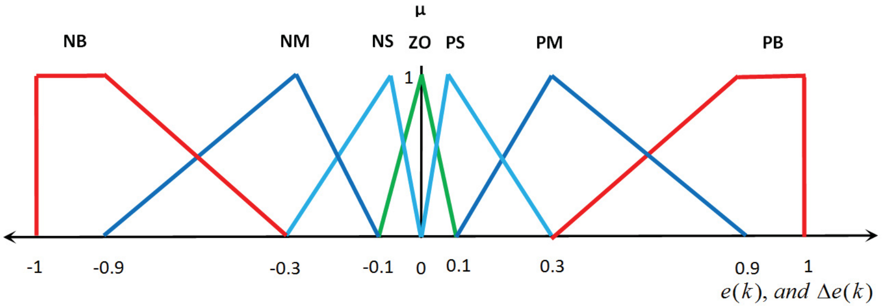


Figure 4.6: Membership function for $e(k)$ and $\Delta e(k)$.

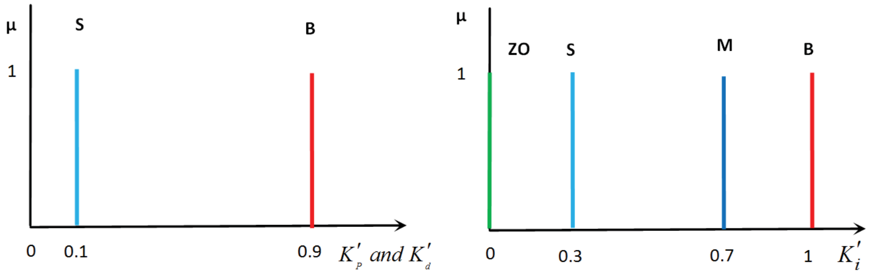


Figure 4.7: Membership function for K_p', K_d' and K_i'.

4.4 Chapter Summary

In this chapter some control algorithms for application to unmanned systems have been presented. First Four controller have been discussed for trajectory tracking control of non-holonomic wheeled mobile robots. Then a stable formation controller for time-varying configurations is presented at the end. Finally a fault tolerant fuzzy gain-scheduled PID controller has been presented for application to quadrotor helicopters. Next chapter will

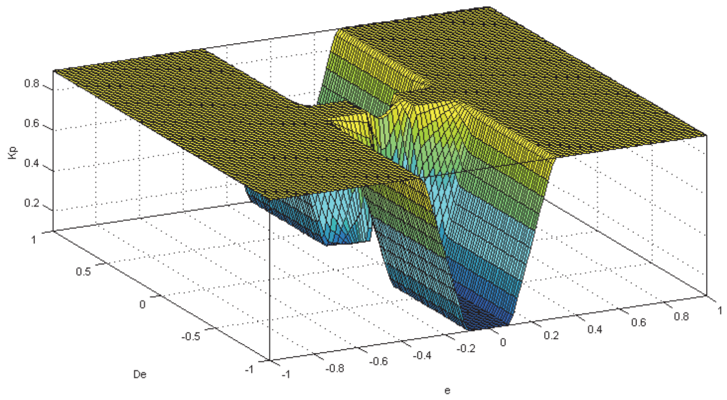


Figure 4.8: Surface for K_p'.

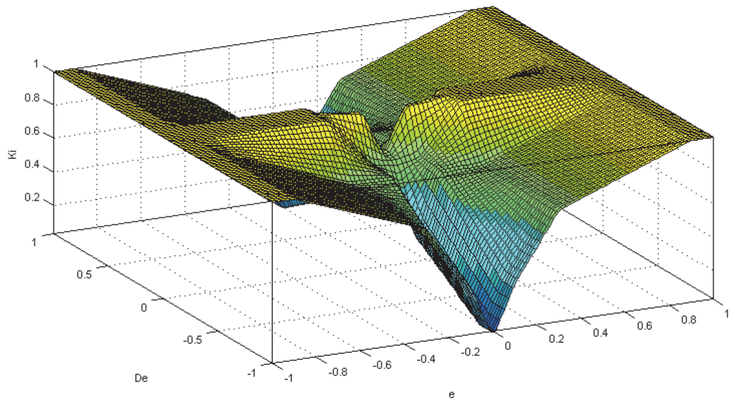


Figure 4.9: Surface for K_i'.

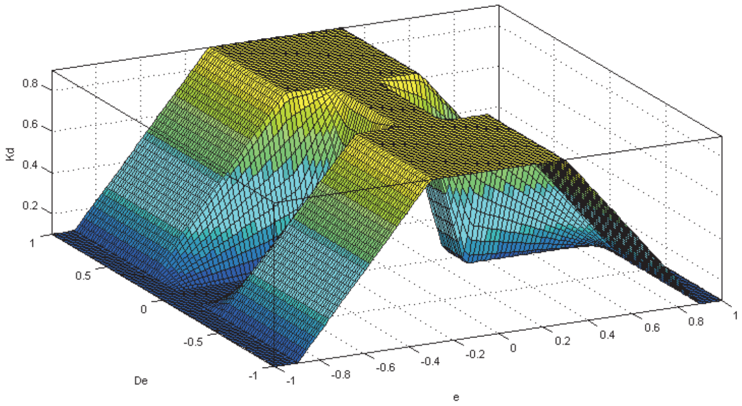


Figure 4.10: Surface for K_d'.

present the simulation and experimental results of applying developed FDD methods in Chapter 3 and Control algorithms which have been developed in this chapter (Chapter 4) into testbeds described in Chapter 2.

Chapter 5

Simulation and Experimental Results with Unmanned Systems

In this chapter simulation and experimental results of developed methods and algorithms are presented. First the effectiveness of proposed trajectory tracking controller is examined via real-world application to the *Qbot* ground vehicle. Then simulation result of the proposed formation controller is presented. The purposed Fault Detection and Diagnosis methods have been experimentally tested on the *Qball-X4* and results will be discussed in this chapter. Then simulation results of fault detection and diagnosis of the momentum wheel fault for *JC2Sat* is presented. Finally experimental result on application of Fault Tolerant Fuzzy Gain-Scheduled PID controller on *Qball-X4* is discussed.

5.1 Experimental Testing Results for Trajectory Tracking Control of Wheeled Mobile Robot

In this section, the effectiveness of the proposed trajectory tracking controller is examined through a real world application to the Qbot test-bed available at Concordia University as shown in Fig. 5.1. The obtained results of the proposed controller are also compared with those of Model Predictive Control (MPC), Linear State tracking Control (LSTC) and Nonlinear State Tracking Control (NSTC) methods respectively. The parameters of the

WMR are adopted from an autonomous three-wheeled mobile robot made by Quanser Inc., as shown in Eq. 5.1 and Fig. 5.1.

$$\begin{aligned} m &= 2kg, & I &= 0.017Kg.m^2, \\ v_{max} &= 0.3m/s, & \omega_{max} &= 1.5rad/s, \\ a_{max} &= 0.2m/s^2, & \dot{\omega}_{max} &= 1rad/s^2 \end{aligned} \tag{5.1}$$

An 8-shaped reference trajectory is chosen as (5.2):

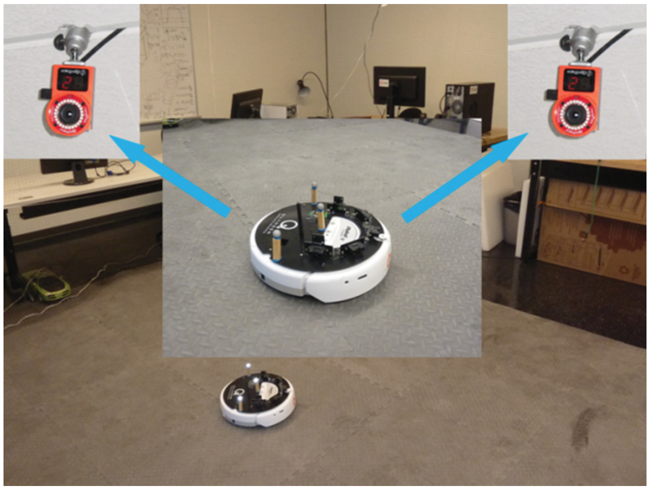


Figure 5.1: The Qbot mobile robot and the OptiTrack camera system set-up at the Networked Autonomous Vehicles Laboratory (NAVL) of Concordia University.

$$\begin{cases} X_r(t) = 1.1 + 0.7sin\left(\frac{2\pi t}{60}\right) & m \\ Y_r(t) = 0.9 + 0.7sin\left(\frac{4\pi t}{60}\right) & m \end{cases} \tag{5.2}$$

The initial conditions of the robot are set as follows:

$$X(0) = 1\ m, \quad Y(0) = -1\ m, \quad \theta(0) = \pi/3\ rad$$

The constant values of the Lyapunov-based Guidance Controller (LGC) proposed in this work and three other methods (MPC, LSTC and NSTC) were tuned as indicated in Table 5.1. In Table 5.1, h is the prediction horizon and T is the time constant of the discrete model of the robot.

Table 5.1: Design constants of LGC, MPC, LSTC and NSTC

LGC	MPC	LSTC	NSTC
$K_v = 0.8$	$h = 8$	$g = 40$	$g = 80$
$K_\omega = 2$	$T = 0.01sec$	$\xi = 0.7$	$\xi = 0.8$
	$R = 10^{-4}\mathbf{I}_{2\times 2}$	$v_{switch} = 0.1$	
	$A_r = \begin{bmatrix} 0.8 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}$	$\omega_n = 2$	
	$Q = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$		

The responses of the proposed algorithm and those of MPC, LSTC and NSTC methods have been compared in Figs. 5.2-5.5. Most of the diagrams presented here for this case indicate that the proposed method has better performance over MPC, LSTC and NSTC techniques. Fig. 5.2 shows the trajectories of four methods. It also shows that LGC moves closer to its desired path compared to other methods.

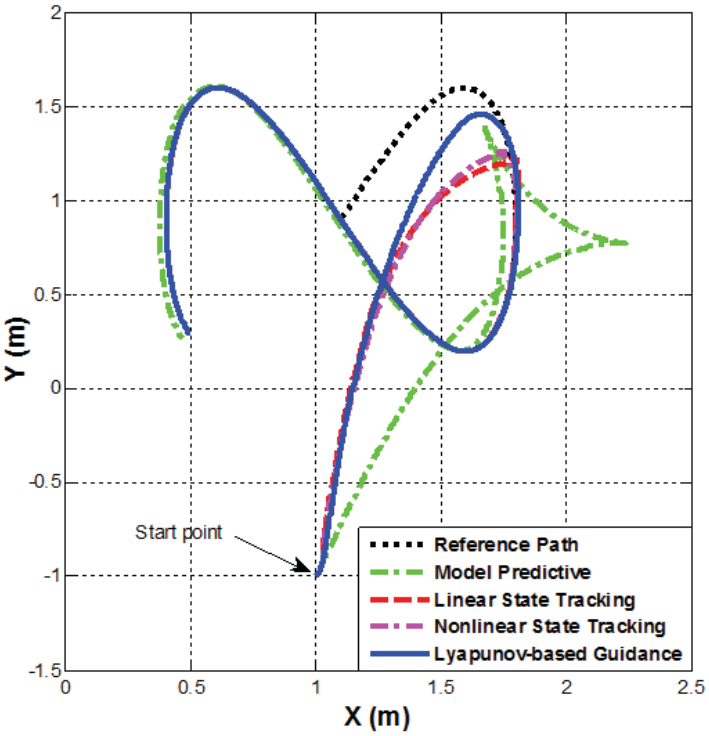


Figure 5.2: Path tracking performance of the LGC versus those of the MPC, LSTC and NSTC techniques in real world implementation based on the Qbot test-bed.

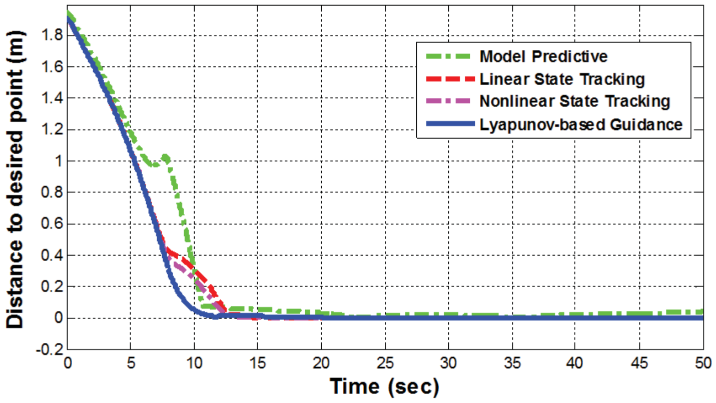


Figure 5.3: Distance between desired and actual positions achieved using LGC versus those of the MPC, LSTC and NSTC techniques in real experimental test on the Qbot.

As it can be seen from Fig. 5.3, the proposed method takes less time (faster tracking) to catch its desired position in reference trajectory compared to other methods.

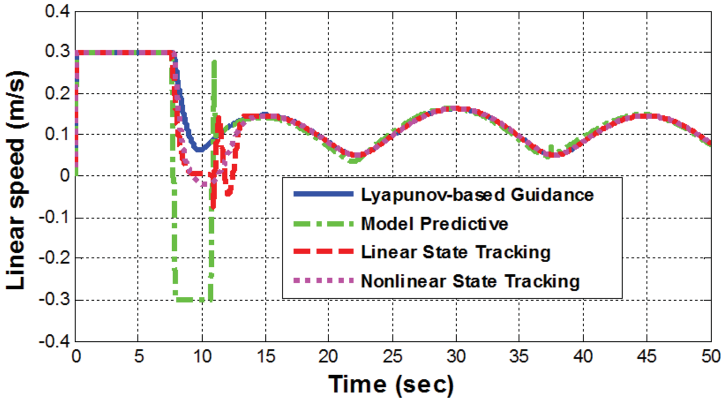


Figure 5.4: Controlled linear velocity of LGC versus those of the MPC, LSTC and NSTC techniques in real experimental test on the Qbot.

Figs. 5.4 and 5.5 show controlled input signals to the robot. As can be seen from Fig. 5.4, the controlled linear velocity is smoother for LGC compared to other methods.

Overall experimental tests through a wheeled mobile robot (Qbot) demonstrated a faster and smoother tracking performance with smoother control inputs requirement by the proposed Lyapunov-based Guidance Controller (LGC) compared to Model Predictive Control (MPC), Linear State Tracking Control (LSTC) and Nonlinear State Tracking Control (NSTC) techniques.

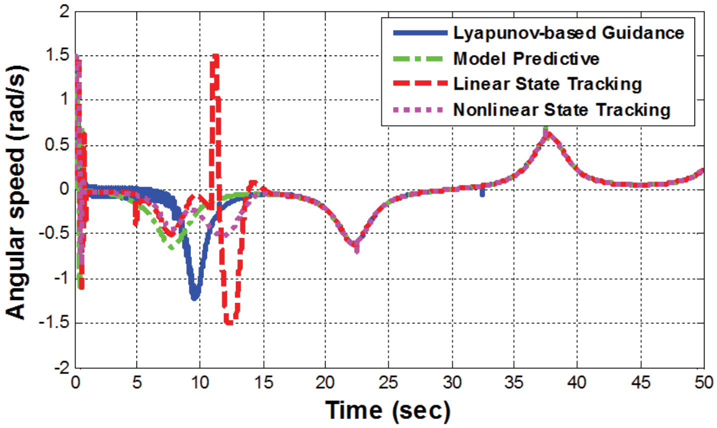


Figure 5.5: Controlled angular velocity of LGC versus those of the MPC, LSTC and NSTC techniques in real experimental test on the Qbot.

5.2 Simulation Results for Time-Varying Formation Control of Multiple Wheeled Mobile Robots

In this section, the effectiveness of the proposed formation controller is examined. For the simulation purposes, following limitations have been taken into account for linear and angular velocities of the robots.

$$v_{max} = 0.3 \;\; m/s, \quad \omega_{max} = 1.5 \;\; rad/s,$$
$$a_{max} = 0.2 \;\; m/s^2, \quad \dot{\omega}_{max} = 1 \;\; rad/s^2$$

The reference trajectory of the leader robot has also been arbitrarily chosen as:

$$v_l = 0.1 \quad m/s;$$
$$\omega_l = 0.01 sin\left(\frac{2\pi t}{400}\right) \quad rad/s;$$

A set of five wheeled mobile robots is used to simulate a time-varying formation control over 500*sec* of simulation time. *Follower 1* and *Follower 2* are made to follow the *leader*. *Follower 3* and *Follower 4* also follow *Follower 1* and *Follower 2* respectively. Fig. 5.6 represents a schematic of each leader-follower pairs. Table 5.2 shows the evolution of desired formation configuration of the group over the time.

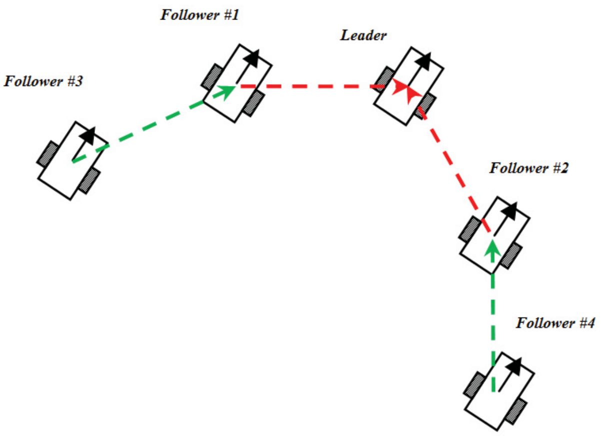


Figure 5.6: The schematic of the robots in formation.

Table 5.2: Formation Configuration

Time (sec)	$0 \sim 150$	$150 \sim 200$	$200 \sim 250$	$150 \sim 350$	$350 \sim 500$
$d_{LF_1}\ \ (m)$	2	$2(1+sin(\frac{2\pi(t-150))}{200})$	4	$2(2-sin(\frac{2\pi(t-250))}{400})$	2
$\varphi_{LF_1}\ \ (rad)$	$\frac{\pi}{6}$	$\frac{\pi}{6}$	$\frac{\pi}{6}$	$\frac{\pi}{6}cos(\frac{2\pi(t-250)}{200})$	$-\frac{\pi}{6}$
$d_{LF_2}\ \ (m)$	2	$2(1+sin(\frac{2\pi(t-150)}{200}))$	4	$2(2-\frac{sin(2\pi(t-250)}{400}))$	2
$\varphi_{LF_2}\ \ (rad)$	$-\frac{\pi}{6}$	$-\frac{\pi}{6}$	$-\frac{\pi}{6}$	$-\frac{\pi}{6}cos(\frac{2\pi(t-250)}{200})$	$\frac{\pi}{6}$
$d_{F_1F_3}\ \ (m)$	0.7	$0.7(1+sin(\frac{2\pi(t-150)}{200}))$	1.4	$0.7(2-sin(\frac{2\pi(t-250)}{400}))$	0.7
$\varphi_{F_1F_3}\ \ (rad)$	$\frac{5\pi}{9}$	$\frac{5\pi}{9}$	$\frac{5\pi}{9}$	$\frac{5\pi}{9}cos(\frac{2\pi(t-250)}{200})$	$-\frac{5\pi}{9}$
$d_{F_2F_4}\ \ (m)$	0.7	$0.7(1+sin(\frac{2\pi(t-150}{200}))$	1.4	$0.7(2-sin(\frac{2\pi(t-250)}{400}))$	0.7
$\varphi_{F_1F_3}\ \ (rad)$	$-\frac{5\pi}{9}$	$-\frac{5\pi}{9}$	$-\frac{5\pi}{9}$	$\frac{5\pi}{9}cos(\frac{2\pi(t-250}{200}))$	$\frac{5\pi}{9}$

The simulation results is presented in Fig. 5.7 to Fig. 5.10.

Fig. 5.7 represents the path of each follower during the simulation. The initial position of the robots are chosen arbitrarily. This figure shows the smooth switching among different configurations.

Fig. 5.8 shows the *virtual line of sight* distance between each follower and its virtual one. This distance can be interpreted as an error factor of the whole formation mission. As it is shown in the figure even during the formation switching the error remains very close to zero. Figures 5.9 and 5.10 show the linear and angular speed of the robots respectively.

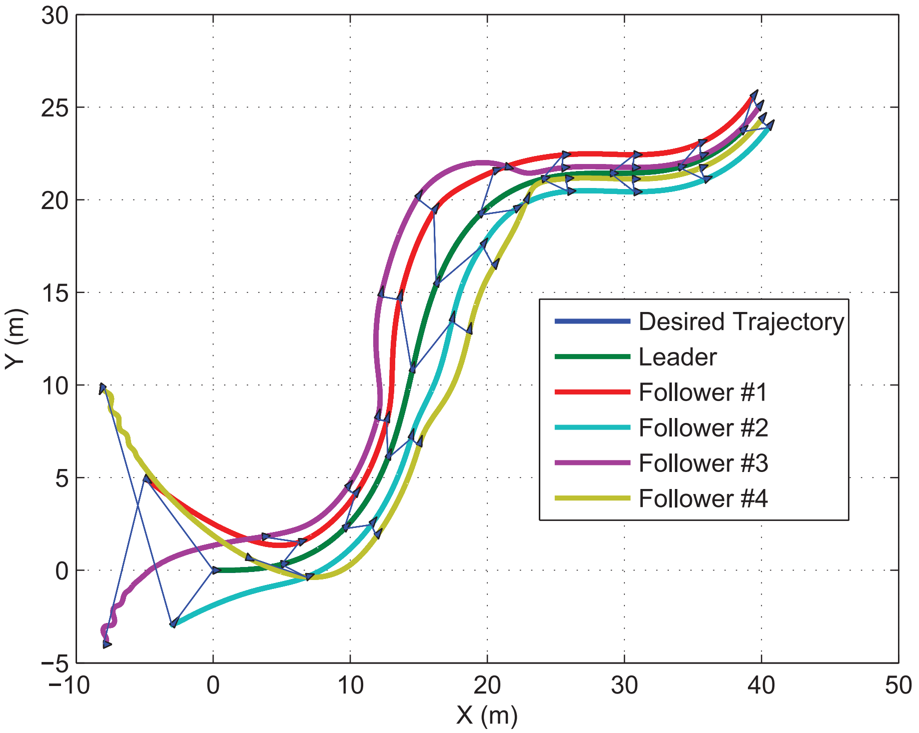


Figure 5.7: Paths of the leader and follower robots.

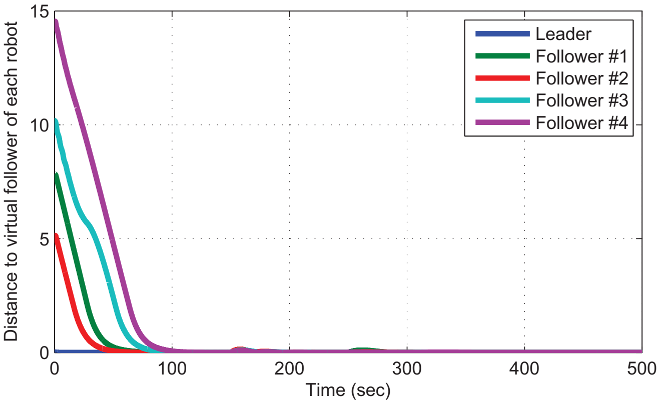


Figure 5.8: Distance between the desired and actual positions (error) of the follower.

5.3 Fault Detection and Diagnosis of the Quadrotor UAV

5.3.1 Actuator Fault Modelling for the Quadrotor Helicopter

Fault detection and diagnosis (FDD) aims to detect abnormal behaviors of a process due to a component failure and eventually isolate the exact location of the failed component and

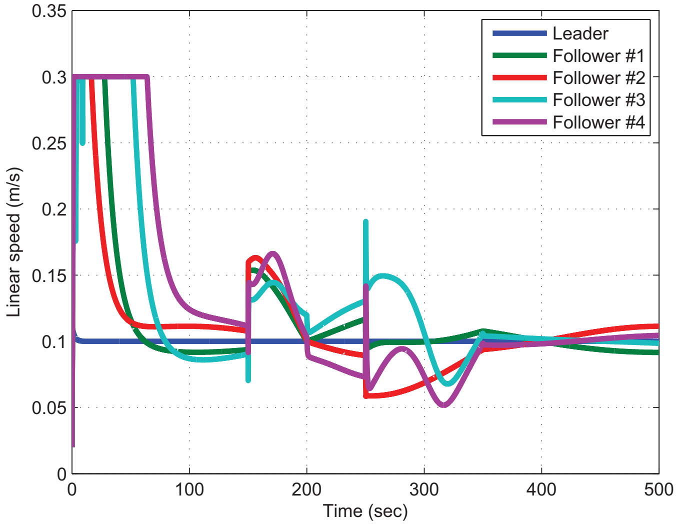


Figure 5.9: Linear speed of the follower robots.

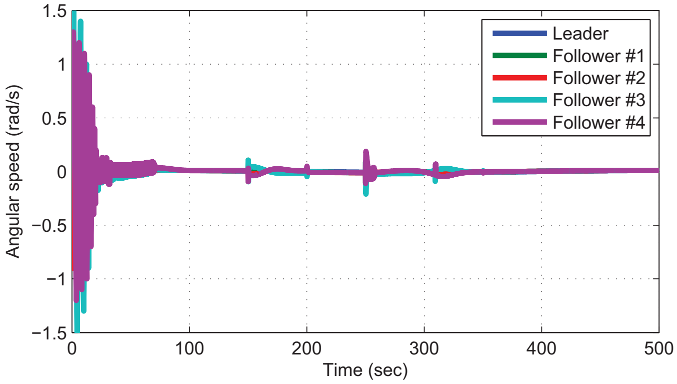


Figure 5.10: Angular speed of the follower robots.

identify the failure type and its severity. This problem has been extensively considered in the literature (see for example [98, 99] and the references therein) since a better knowledge of the failure location, type and amplitude greatly helps in minimizing the fault effects on the process behavior. This is particularly important for safe operation and/or fault tolerant control of safety-critical systems, such as aircrafts, spacecrafts, nuclear power plants, and chemical plants processing hazardous materials where the consequences of a minor fault in a system component can be catastrophic.

This section considers the FDD problem for the quadrotor helicopter in the presence of actuator faults. Such faults can be modeled as follows:

$$u_i^f = (1-\gamma_i)\,u_i \leq (1-\gamma_i)\,u_{max} \;\;;\;\; (i = 1,\cdots,4) \tag{5.3}$$

where γ_i represents the loss of effectiveness in the i^{th} rotor. $\gamma_i = 0$ denotes a healthy rotor, $\gamma_i = 1$ denotes a complete loss of the i^{th} rotor and $0 < \gamma_i < 1$ represents a partial loss of control effectiveness. During the mission and after the occurrence of a fault in one of the UAV rotors, two cases can be distinguished: a) the damaged UAV cannot recover and the system is lost; b) the damaged UAV recovers and maintains its stability. It can be shown from (2.6) and (2.9) that the required PWM inputs to keep a quadrotor UAV in hovering flight condition (i.e. for $x = 0$, $y = 0$ and $\psi = 0$) are $u_i^* = mg/4K$ for $i = 1,...,4$. Thus, a damaged quadrotor can always maintain its altitude if:

$$u_i^* = \frac{mg}{4K} \leq (1-\gamma_i)\,u_{max} \;\;;\;\; i = 1,...,4 \tag{5.4}$$

or

$$\gamma_i \leq 1 - \frac{mg}{4Ku_{max}} \;\;;\;\; i = 1,...,4 \tag{5.5}$$

The system parameters along with their numerical values are given in Table 5.3. Substituting *Qball-X4* numerical values in 5.5, the maximum tolerable loss of control effectiveness can be calculated as:

$$\gamma = 1 - \frac{1.42 \times 0.81}{4 \times 120 \times 0.05} \simeq 0.4 \tag{5.6}$$

Table 5.3: Qball-X4 Parameters

Parameter	Description	Value
K	Thrust gain	120
ω	Motor bandwidth	15 rad/sec
L	Distance from motor to CG	0.2 m
K_ψ	Thrust-to-moment gain	0.023
m	Mass	1.42 kg
g	Gravity	9.81 m/s^2
$J_1;J_2;J_3$	Moments of inertia	0.03; 0.03; 0.04 $kg.m^2$

5.3.2 Linearized Model of the Quadrotor UAV

The application of the TSKF, ATSKF and IMM to the quadrotor UAV starts by deriving a linearized state-space model. Combining (2.6) and (2.9) yields to the linearized state-space model:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{z} \\ \ddot{z} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\phi} \\ \ddot{\phi} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \\ \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ K/m & K/m & K/m & K/m \\ 0 & 0 & 0 & 0 \\ KL/J_1 & -KL/J_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & KL/J_2 & -KL/J_2 \\ 0 & 0 & 0 & 0 \\ KK_\psi/J_3 & KK_\psi/J_3 & -KK_\psi/J_3 & -KK_\psi/J_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} g$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \vdots \\ \psi \\ \dot{\psi} \end{bmatrix}$$

5.3.3 Fault Modeling Using Multiple Models

Sixteen different model have been used as Multiple-Models inside the FDD block. One represents healthy mode, and other fifteen models represent faulty modes. In real fly, due to actuator saturation, the Qball-X4 cannot tolerate more than 40 $\sim$ 45 percent loss of effectiveness in each actuator. In other words, if the fault goes beyond 40 $\sim$ 45 percent, Qball-X4 cannot be recovered and will crash. Tacking this fact into consideration, faulty models are designed based on 40 percent loss of actuator's effectiveness. Table 5.4 shows description of models used as multiple models. For example Model 7 is used to show the case in which Actuator 1 and 3 perform normally, while Actuator 2 and 4 lose 40 percent of their effectiveness.

Table 5.4: Description of models used in IMM, H: Actuator is Healthy, F: Actuator is Faulty

Rotor ╲ Model	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Actuator 1	H	F	H	H	H	F	H	F	H	H	F	H	F	F	F	F
Actuator 2	H	H	F	H	H	H	F	F	F	H	H	F	H	F	F	F
Actuator 3	H	H	H	F	H	F	H	H	F	F	H	F	F	H	F	F
Actuator 4	H	H	H	H	F	H	F	H	H	F	F	F	F	F	H	F

5.3.4 Implementation of TSKF, ATSKF and IMM on Qball-X4

Experimental Results

The TSKF, ATSKF and IMM have been tested on the Qball-X4 testbed of the NAV lab (Figure 5.11). Several experiments are carried out in the presence of actuator faults, flight data is collected and then the filters are applied to the data off-board to diagnose faults. The experiments are taking place indoor in the absence of GPS signals and thus the OptiTrack camera system from NaturalPoint is employed to provide the system position in the 3D space. In all experiments, the system is required to hover at an altitude of 1 m and the faults are taking place at time instant $t = 15$ s.

Figure 5.11: The NAV Lab of Concordia University.

The covariance matrices for the implementation on the Qball-X4 UAV are set to:

$$
\begin{aligned}
Q^x &= \begin{bmatrix} 10^{-6} \times I_{6\times6} & 0_{6\times6} \\ 0_{6\times6} & 10^{-12} \times I_{6\times6} \end{bmatrix} \\
R &= \begin{bmatrix} 10^{-6} \times I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 10^{-12} \times I_{3\times3} \end{bmatrix} \\
Q^\gamma &= 10^{-1} \times I_{4\times4}
\end{aligned}
$$

The transition probability used is given as:

$$\pi_{16\times16} = \begin{bmatrix} 0.995 & 3\times10^{-4} & 3\times10^{-4} & \dots & 3\times10^{-4} \\ 0.005 & 0.995 & 0 & \dots & 0 \\ 0.005 & 0 & 0.995 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0.005 & 0 & \dots & 0 & 0.995 \end{bmatrix}$$

Single fault scenario: control effectiveness loss in the third motor

In the first scenario, a loss of control effectiveness of 45% is simulated in the third motor (i.e. $\gamma_3 = 0.45$). As can be seen in Figure 5.12, this fault does not affect the system along the x-direction but it results in loss of altitude of 20 *cm* and a deviation of 80 *cm* from the desired position along the y-direction. Due to the presence of a controller, the system recovers and goes back to the desired hover position. Figure 5.13 shows the behavior of system in the 3D space upon fault injection. The PWM inputs to the four motors are illustrated in Figure 5.14. It is clear that before fault (up to 15 seconds), all the four PWM inputs are almost the same. After fault injection in the third motor, the baseline controller reacts by automatically increasing the third PWM input to compensate the occurred fault. The estimations of the actuator fault provided by the TSKF and the ATSKF are given in Figures 5.15 and 5.16 respectively. The experimental application shows a fast and precise estimation of the fault amplitude despite model uncertainties. The estimates of γ_1, γ_2 and γ_3 remain close to zero whereas that of γ_3 converges to a value close to 0.45. The experimental results using IMM are show in Figures 5.17 and 5.18. As it can be seen from Figure 5.17 the probability of the first model (represents healthy model) is the highest (around 0.9). In the 15 second when the fault is occurred probability of the forth model (represents fault in third actuator) increases (around 1). Figure 5.18 also show the effective model over the time.

Simultaneous faults scenario: control effectiveness loss in all motors

In the second scenario, a loss of control effectiveness of 40% is simulated in all motors (i.e. $\gamma_i = 0.4$ for $i = 1, \dots, 4$). Unlike the previous case, this fault does not affect the system position in the x and y directions but results in a larger loss in altitude due to the feature of the faults, where the system drops to 0.4 *m* (see Figure 5.19). Figure 5.20 shows the system's behavior in the 3D space upon fault injection. The PWM inputs to the four motors are illustrated in Figure 5.21. Up to 15 seconds and before fault injection, all the four PWM inputs are almost the same. After fault injection, the baseline controller automatically increases

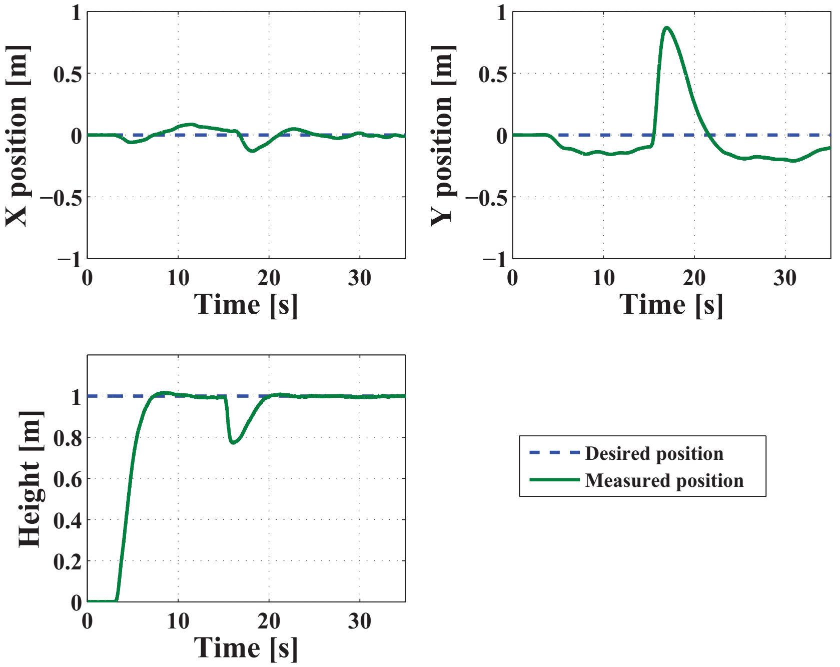


Figure 5.12: System position along *x*, *y* and *z* directions for single fault scenario.

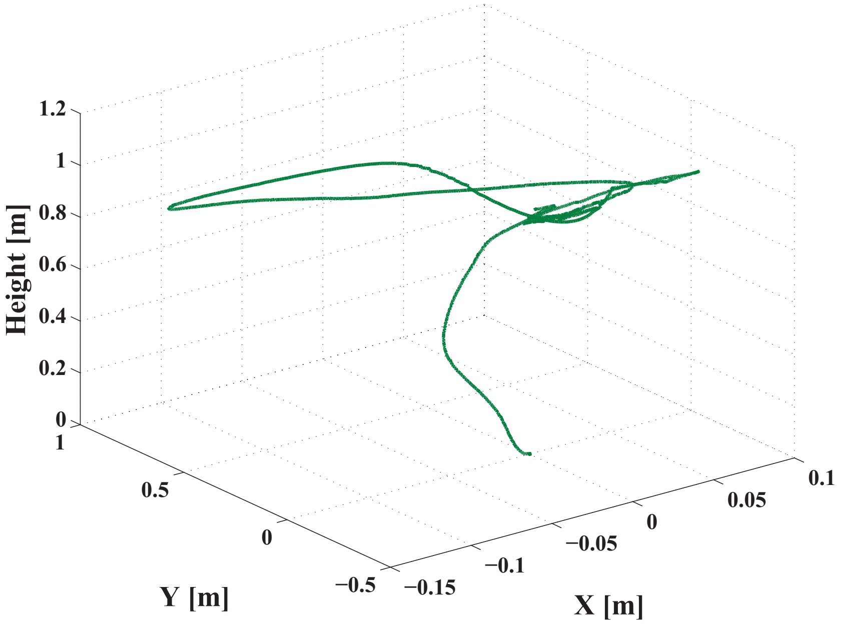


Figure 5.13: System's 3D position for single fault scenario.

the PWM inputs to compensate the occurred faults. The estimation of the actuator faults using TSKF and ATSKF is given in Figures 5.22 and 5.23 respectively. Once again, the experimental application shows a fast and good estimation of all fault amplitudes despite model uncertainties. The experimental results using IMM are shown in Figures 5.24 and

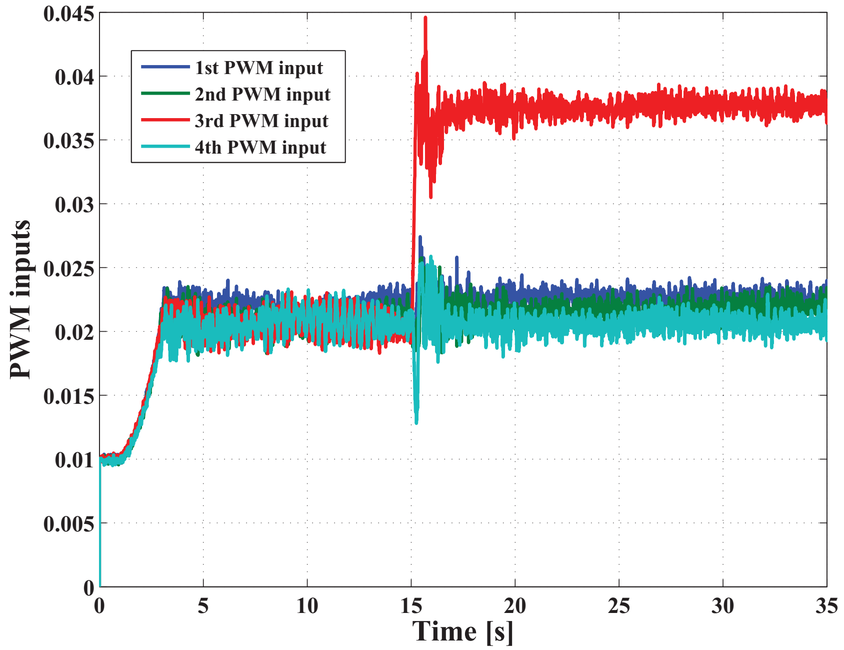


Figure 5.14: The PWM inputs generated by the controller for single fault scenario.

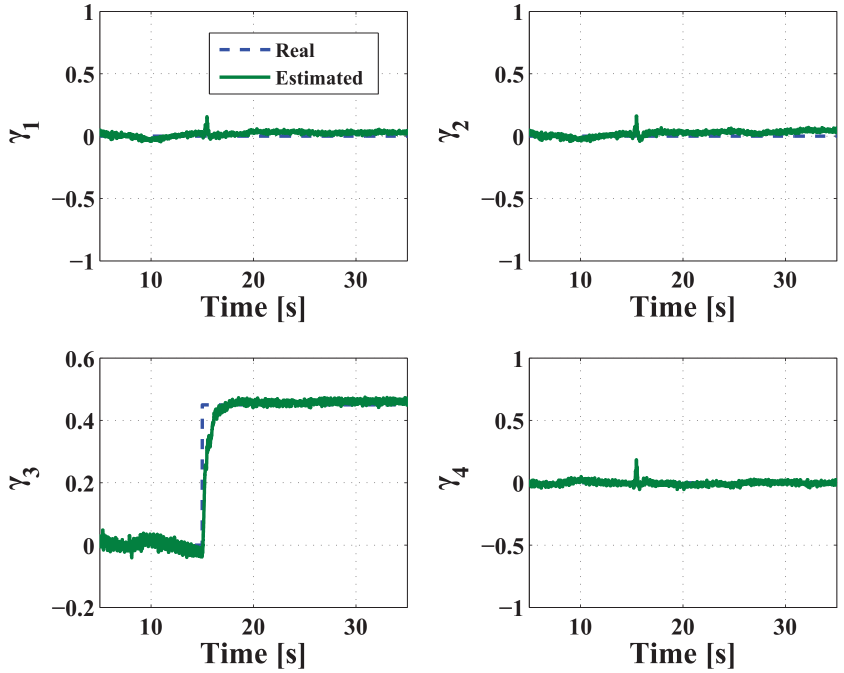


Figure 5.15: TSKF control effectiveness factors estimation for single fault scenario.

5.25. The mode probabilities of all sixteen models is given in Figure 5.24. As it can be seen from Figure 5.24 the probability of the first model (represents healthy model) is the highest (around 0.9). In the 15 second when the fault is occurred probability of the first model decreases (around 0.3) and the probability of the sixteenth model (represents fault

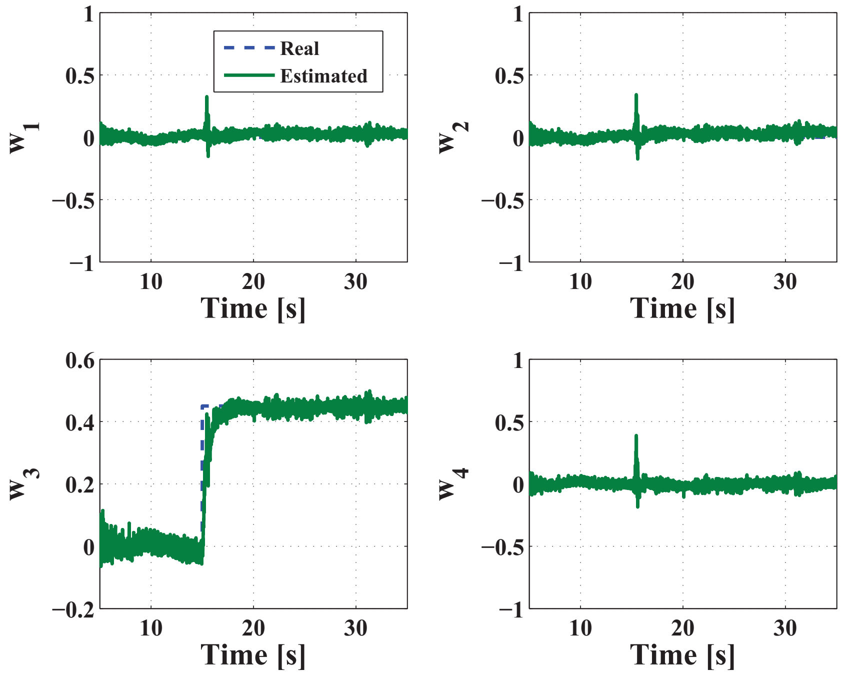


Figure 5.16: ATSKF control effectiveness factors estimation for single fault scenario.

in all actuator) increases (around 0.7). Figure 5.25 also show the effective model over the time.

The application of TSKF, ATSKF and IMM to the Qball-X4 has shown to be effective

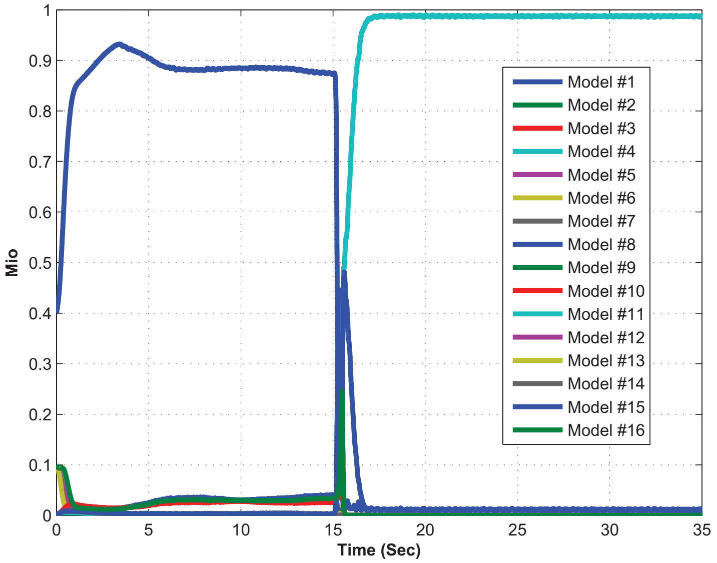


Figure 5.17: Mode probabilities for single fault scenario.

in estimating actuator faults. A comparison of the obtained results does not allow to draw an obvious superiority of one approach over the others. Moreover, fine tuning the filter parameters can always improve the results, making them dependent on the gains selection. The results show that ATSKF and TSKF were very close to each other in terms of fault estimation time and IMM reacted a little bit slower than TSKF and ATSKF.

Due to the limited calculation capabilities of the on-board microcomputer, the computational complexity of the filters is an important factor to investigate. In the current situation, the Gumstix embedded computer runs with a frequency of 200 Hz which is equivalent to a sample time of 0.005 seconds. In real-time systems, the timing behavior is an important property of each task. It has to be guaranteed that the execution of a task does not take longer than a specified amount of time. Thus, a knowledge about the maximum execution time of programs is of utmost importance [100]. In terms of computational complexity, IMM needs more powerful processor and cannot be implemented on board. TSKF and ATSKF both seem work well, although ATSKF puts more load on the system compared to TSKF.

For comparison purpose, TSKF and ATSKF are implemented on a desktop computer running Intel Core i5 CPU with 2.67 GHz processing speed and 2.99 GB of RAM. Moreover, 100 different runs are carried out to reduce possible effects of programs running in the background. Table 5.5 gives a quantitative comparison of the mean time (in seconds)

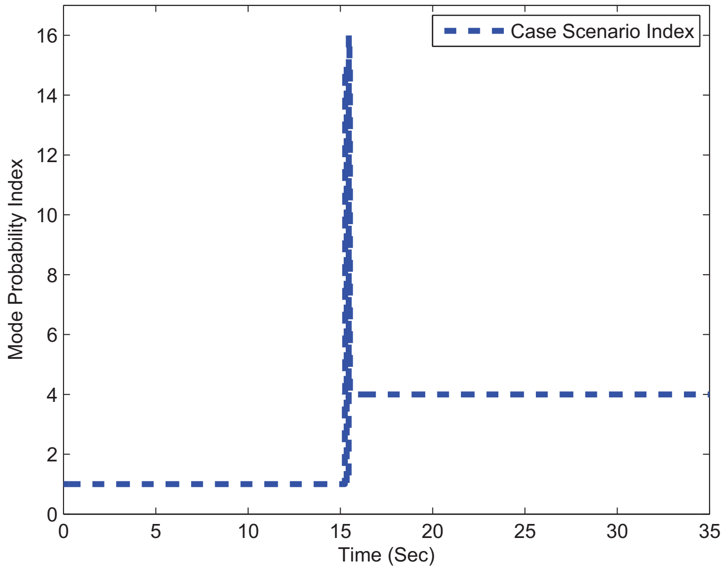


Figure 5.18: Effective model index for single fault scenario.

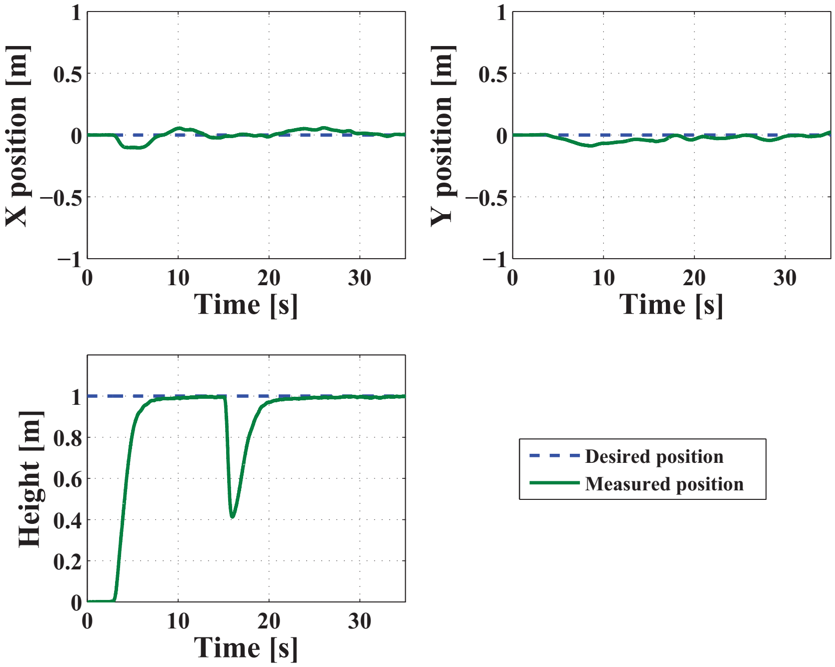


Figure 5.19: System position along *x*, *y* and *z* directions for simultaneous faults scenario.

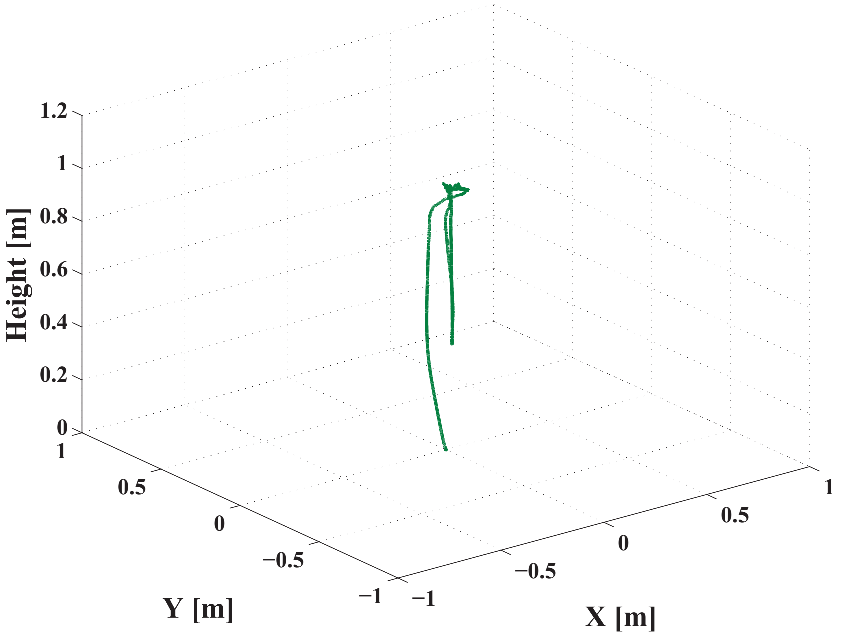


Figure 5.20: System's 3D position for simultaneous fault scenario.

taken by each filter to execute. One can see that the ATSKF is twice slower to execute than the TSKF. Figure 5.26 shows the execution times for each of the filters in the single fault case. The left hand side of the figure illustrates the mean of the execution times for the 100 runs. The right hand side shows one of the 100 runs and the execution time for each sample

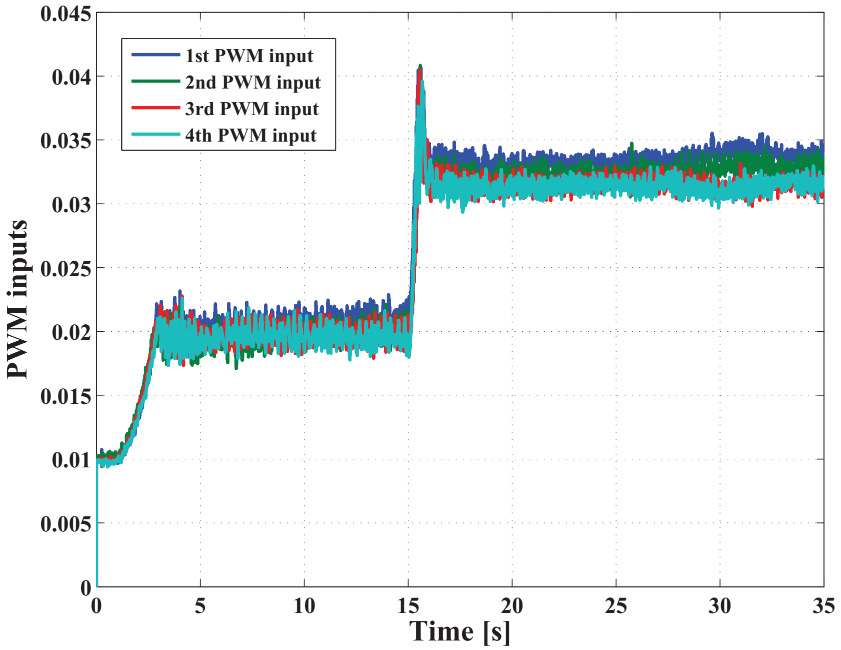


Figure 5.21: The PWM inputs generated by the controller for simultaneous fault scenario.

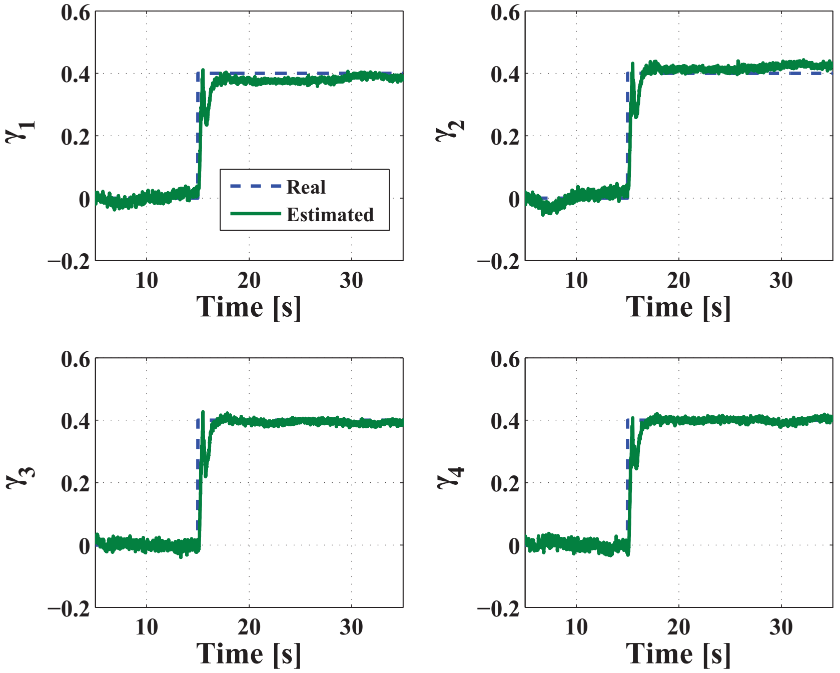


Figure 5.22: TSKF control effectiveness factors estimation for simultaneous fault scenario.

time.

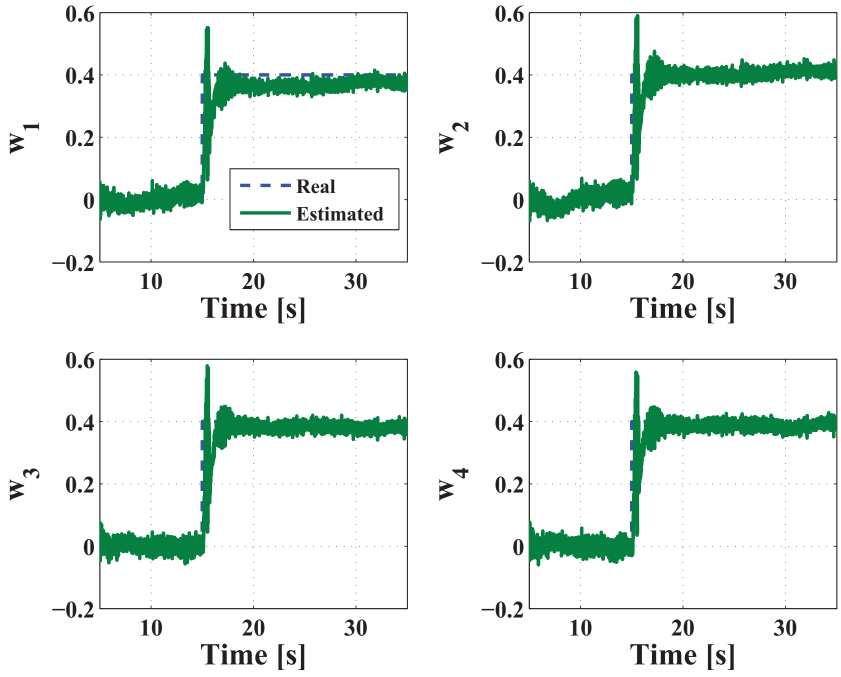


Figure 5.23: ATSKF control effectiveness factors estimation for simultaneous fault scenario.

Table 5.5: Mean of execution times (in seconds) for 100 runs.

Fault case	TSKF	ATSKF
Single fault	3.4478×10^{-4}	7.2427×10^{-4}
Simultaneous faults	3.5102×10^{-4}	7.3627×10^{-4}

5.4 Fault Detection and Diagnosis of the Momentum Wheel Fault in JC2Sat

In this section simulation results of using Interacting Multiple Model for the purpose of fault detection and diagnosis of different probable fault scenarios on momentum wheel of JC2Sat is presented and discussed.

5.4.1 Fault Modeling for JC2Sat

As shown in Fig. 5.27, the FDD scheme receives inputs from the controller output to be sent to actuators and output of the spacecraft. In FDD block, first a residual is generated between the true signal and the estimated signal. If the residual is zero it is understood that there is no fault and if the residual is non-zero then a fault has occurred in an actuator. Knowing that there is a fault in the system the next step is to estimate the faulty parameters.

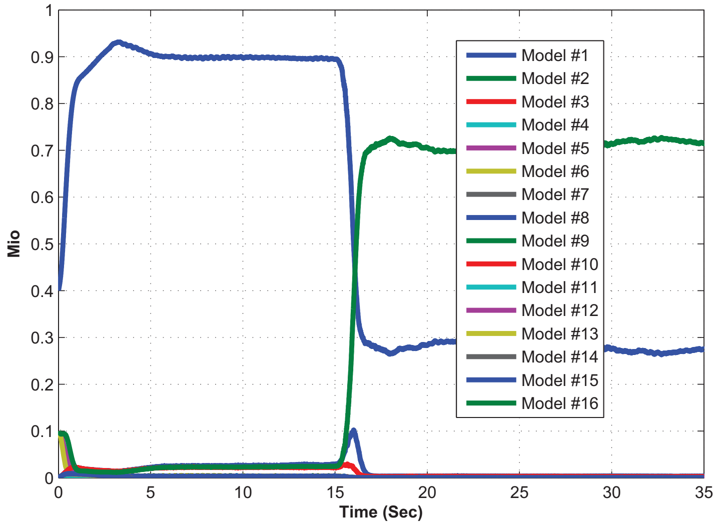


Figure 5.24: Mode probabilities for simultaneous fault scenario.

To estimate the faulty parameters in the actuator, the actuator model (Momentum Wheel) has been presented in subsection 2.3.3.

Based on the state-space model Eq. 2.17 and Eq. 2.18, a FDD scheme can be designed. Before proceeding to the FDD algorithm, potential faults in momentum wheels are discussed first. One of the most probable fault scenarios in momentum wheels is the change in friction. Usually when momentum wheels are aged, wear of bearings and coefficient of viscous friction gradually increase when the inner surface of bearings corrode. Increase in friction will also decrease actuation power of momentum wheel, which may affect overall performance of the controlled system and in a worse situation may lead to instability of the system. The second potential fault in momentum wheel is the change in electrical characteristics of the motor. The electrical resistance may face abrupt changes due to tear of winding in armature of motor. The faults in the electrical characteristics can be modelled as change in the motor torque coefficient K_M. Correct estimation of these parameters will help to reconfigure control structure. This motivates to estimate both the states and parameters. On a general point of view, this problem may be addressed as a dual estimation technique to simultaneously estimate both the states and the parameters of the system. These trigger the design of a FDD algorithm to estimate coefficient of friction and current coefficient in a model of momentum wheel along with the state of the process. An IMM-based FDD strategy is used for this purpose and is discussed in the next section. As discussed earlier, an Adaptive Two-Stage Kalman Filter (ATSKF) (Amoozgar et al., 2011)

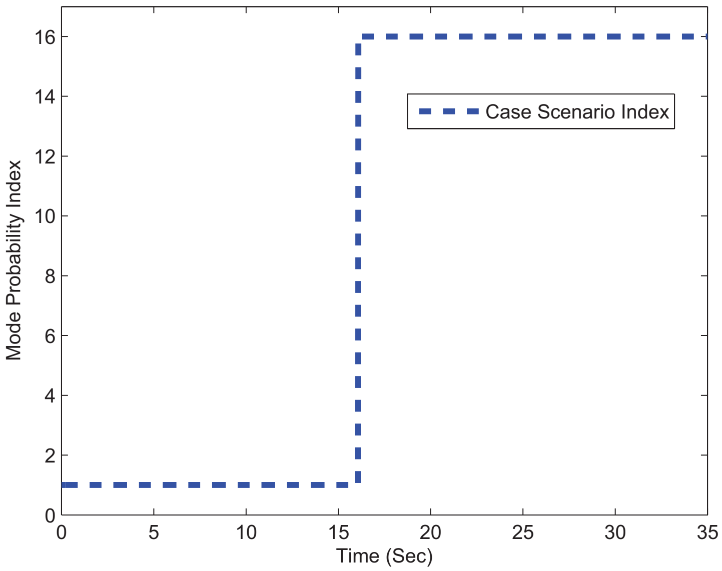


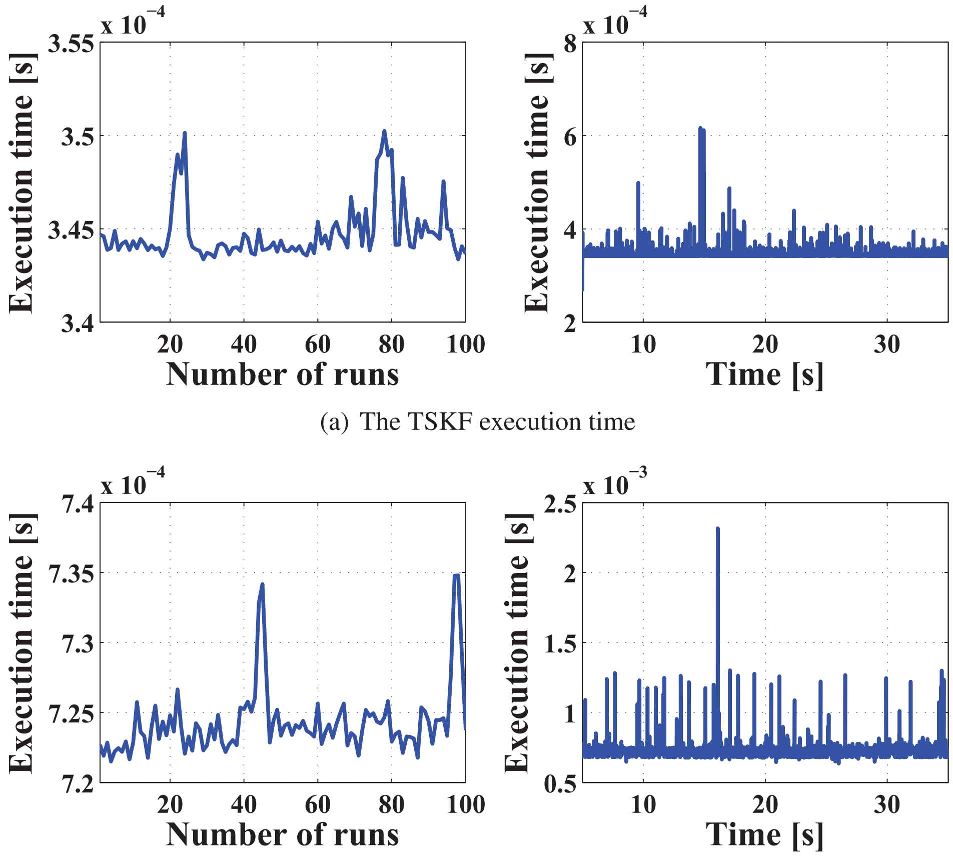
Figure 5.25: Effective model index for simultaneous fault scenario.

has been designed for such a problem. Although ATSKF was designed successfully for this mission, there was a limitation about the detection of simultaneous faults. In other words, the pitch control channel of JC2Sat-FF is controlled just with one commanded input, while to detect simultaneous faults in the system, ATSKF needs at least two commanded inputs. Therefore based on the above limitations an IMM-based FDD strategy is used to overcome this shortage. More design details are discussed in the next section.

5.4.2 Simulation Results of IMM on JC2Sat

In this part simulation results in the MATLAB/Simulink environment are presented. The simulation parameters used are given in Table 5.6. First, spacecraft dynamics along with a controller is simulated for fault-free case. Since the momentum wheel is used only in the pitch axis, only simulation results for pitch axis is shown. However, it should be noted that algorithms were tested in entire 6 DOF nonlinear model. In addition, as the mission is to demonstrate the feasibility of maintaining along-track spacecraft formation by using only differential atmospheric drag control between the two satellites, the change in orientation of one satellite changes the orientation of the other. Keeping this in mind, simulations for one satellite are performed.

The initial conditions for the simulations are described below. As mentioned earlier, four multiple models, one healthy model and three faulty models, have been modelled for



(a) The TSKF execution time

(b) The ATSKF execution time

Figure 5.26: Execution times of TSKF and ATSKF in the single fault case.

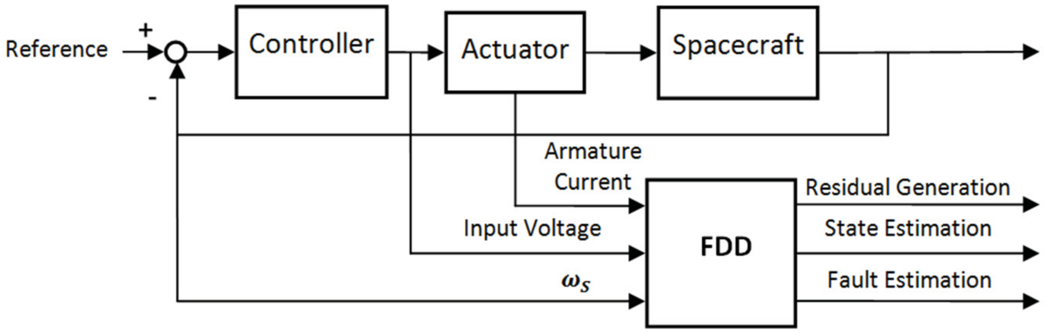


Figure 5.27: Schematic of spacecraft system.

FDD purpose. The three faults considered are 1) fault in current coefficient, 2) fault in friction, and 3) a simultaneous fault mode induced by both current coefficient and friction. These four models are given as:

$$\hat{x}_1 = \hat{x}_2 = \hat{x}_3 = \hat{x}_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad P_1 = P_2 = P_3 = P_4 = 10 \times \mathbf{I}_{2\times 2}$$

Table 5.6: Simulation parameters of the JC2Sat-FF mission

I_x	$0.5845 kg.m^2$
$I_s = I_y$	$0.5584 kg.m^2$
I_z	$0.5320 kg.m^2$
I_w	$8.8147 \times 10^{-5} kg.m^2$
R_m	2Ω
K_m	$1 \times 10^{-3} N.m/A$
B	$9.8 \times 10^{-4} N.m.s$
K_{BEMF}	$1 \times 10^{-3} V.s$

$$Q_1 = Q_2 = Q_3 = Q_4 = \begin{bmatrix} 1 \times 10^2 & 0 \\ 0 & 1 \times 10^{-7} \end{bmatrix} R_1 = R_2 = R_3 = R_4 = \begin{bmatrix} 1 \times 10^1 & 0 \\ 0 & 1 \times 10^{-7} \end{bmatrix}$$

The transition probability used in these simulation results is given as:

$$\pi = \begin{bmatrix} 1 - 3 \times 10^{-4} & 1 \times 10^{-4} & 1 \times 10^{-4} & 1 \times 10^{-4} \\ 2 \times 10^{-4} & 1 - 2 \times 10^{-4} & 0 & 0 \\ 2 \times 10^{-4} & 0 & 1 - 2 \times 10^{-4} & 0 \\ 2 \times 10^{-4} & 0 & 0 & 1 - 2 \times 10^{-4} \end{bmatrix}$$

The simulation results are provided for four scenarios. The scenarios consider all the possible faults, single fault, and multiple faults occurring consequently and simultaneously. First the simulations start with a healthy mode and at 200 seconds a fault in current coefficient is initiated. This is followed by a fault in friction at 400 seconds. The faults are injected consequently in a single run. To further validate the performance of the algorithm, a simultaneous fault is injected at 600 seconds, a fault in friction and fault in current coefficient.

In Fig. 5.28, the commanded voltage is shown. The commanded voltage is the input to the momentum wheel. It is obtained from the controller. The controller is based on the LQR design. As shown in the figure, the commanded voltage changes in lieu of the fault injection at 200, 400 and 600 seconds respectively.

In Fig. 5.29, the pitch angle response is shown. The controller tries to stabilize the spacecraft in pitch axis at about the first 50-75 seconds. This is followed until there is a fault injection in the current coefficient at 200 seconds. As noticed, the response is delayed by a couple of seconds before the effect is noticed. At 400 seconds, a fault in friction is injected, as can be seen in the zoomed figure, it is noticed that there is indeed a change in the behaviour which is larger than the current coefficient fault. At 600 seconds, simultaneous

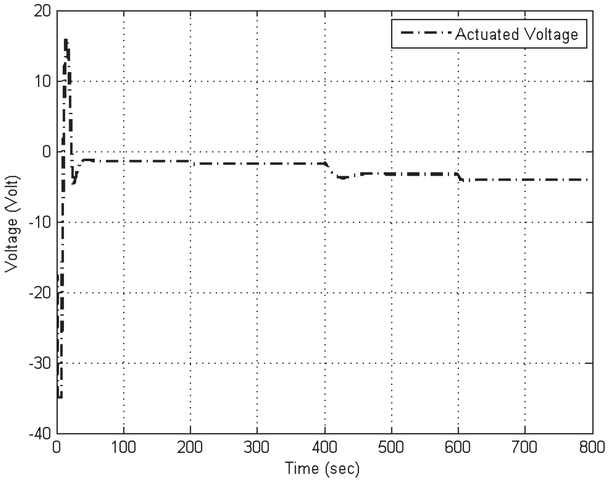


Figure 5.28: Commanded voltage.

fault is injected and a small change in the behaviour can be observed. The reason behind the small changes is because the simulations are performed in a closed-loop environment and therefore the controller is robust enough to stabilize the spacecraft for small faults, which also makes the detection and isolation of faulty mode more challenge and difficulty. The future work of this paper would be to identify the severe faults when the controller cannot stabilize the spacecraft any more.

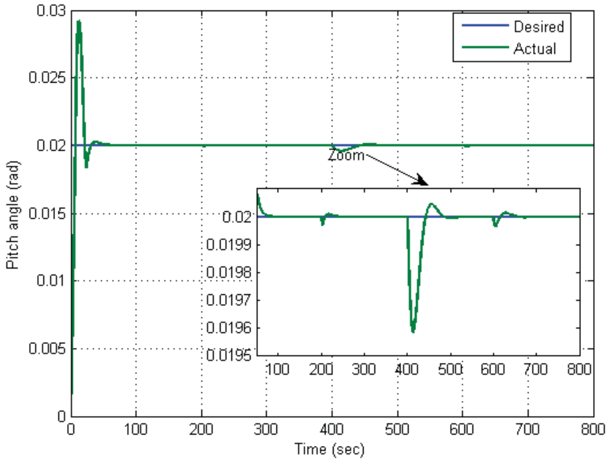


Figure 5.29: Pitch angle for the commanded torque voltage.

Figure 5.30 shows the mode probabilities for the above mentioned faults occurred at 200, 400 and 600 seconds, respectively. As noticed there are 4 scenarios shown in the figure. The first scenario refers to the healthy mode. The second scenario is at 200 seconds, which involves fault in current coefficient. Fault in friction is injected at 400 seconds as the third scenario and finally the fourth scenario is the case when both faults are injected simultaneously. The probability of each mode varies between 0 and 1. From the figure it is evident that which mode is active. Figure 5.31 shows the probability index of all the four modes. As noticed there is not much deviation in between each mode which explains that there is no incorrect information about mode estimation.

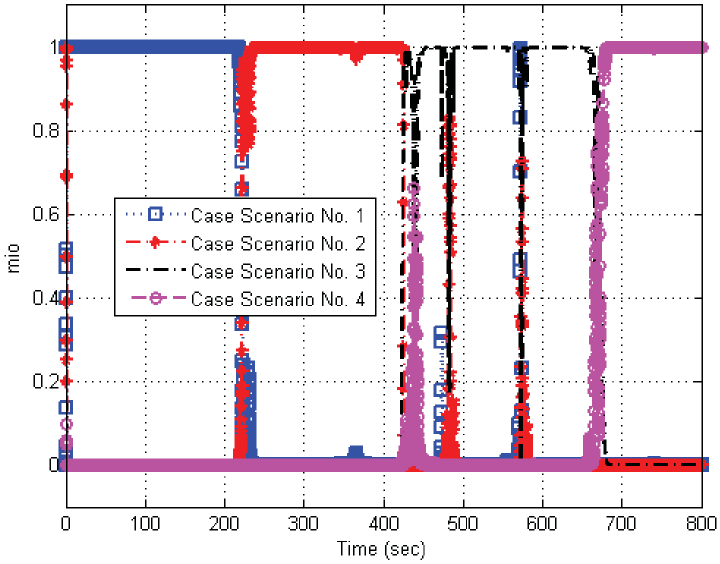


Figure 5.30: Mode probabilities of healthy and faulty modes.

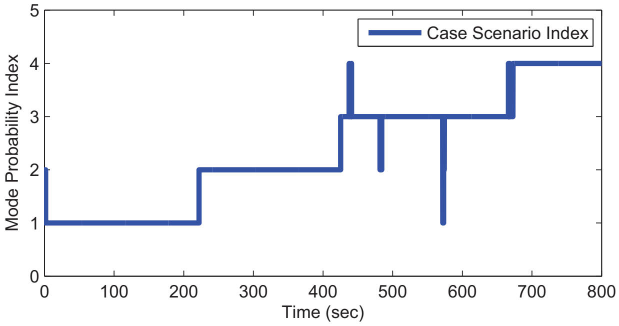


Figure 5.31: Valid mode of operation.

5.5 Experimental Testing Results of Fault Tolerant Fuzzy Gain Scheduling PID Control on Qball-X4

The fuzzy PID controller proposed in Section 4.3 has been experimentally tested on the Qball-X4 testbed. The controller is built using Matlab/Simulink and downloaded on the

Gumstix emdedded computer to be run on-board with a frequency of 200 Hz. The experiments are taking place indoor in the absence of GPS signals and thus the OptiTrack camera system from NaturalPoint is employed to provide the system position in the 3D space. In all experiments, the system is required to hover at an altitude of 1 m and the faults are taking place at time instant $t = 20$ s.

5.5.1 First Fault Scenario

In the first fault scenario, it is assumed that a loss of control effectiveness of 15% is taking place in the four motors. This kind of fault results in a loss of altitude and does not really produce significant movement along the x or y directions. The gains of the conventional PID for the height control are K_p = 0.0122, K_i = 0.0079, and K_d = 0.0093. The predefined ranges of K_p, K_i, and K_d for the fuzzy gain-scheduled PID in the height control are $K_{p,\min}$ = 0.010, $K_{p,\max} = 0.015$, $K_{i,\min} = 0.007$, $K_{i,\max} = 0.010$, $K_{d,\min} = 0.0085$, and $K_{d,\max} = 0.0095$. Figure 5.32 shows a comparison between the conventional and the fuzzy adaptive PID controllers for the height holding flight. It is clear that the fuzzy adaptive PID controller reduces the fault effect on the system by reacting faster and returning the system quicker to its hovering position.

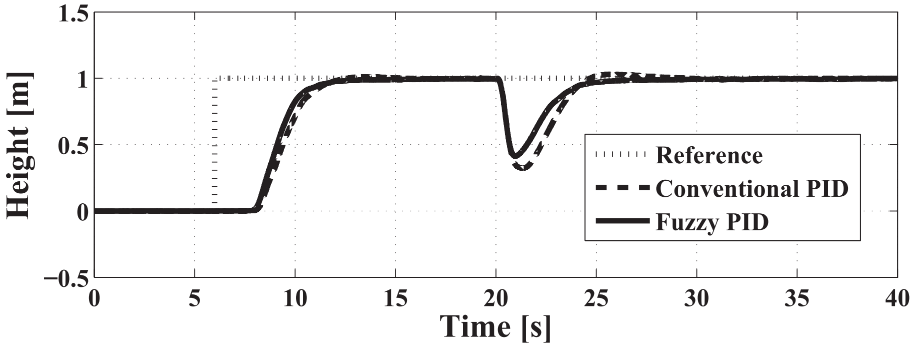


Figure 5.32: Comparison between conventional and fuzzy PID.

The time evolutions of the fuzzy PID gains are illustrated in Figure 5.33. Unlike those of the conventional PID, the fuzzy gains are time-varying to adapt to uncertainties, disturbances and faults as can be clearly seen at $t = 20$ s.

It can be seen in Figure 5.33 that after the fault occurs, K_p decreases to avoid system overshoot due to increase in tracking error. The derivative gain K_d remains fixed with a high value to make a fast response to sudden changes in tracking error. When the system stops descending (loosing altitude) K_d decreases to let the system recovers faster and goes

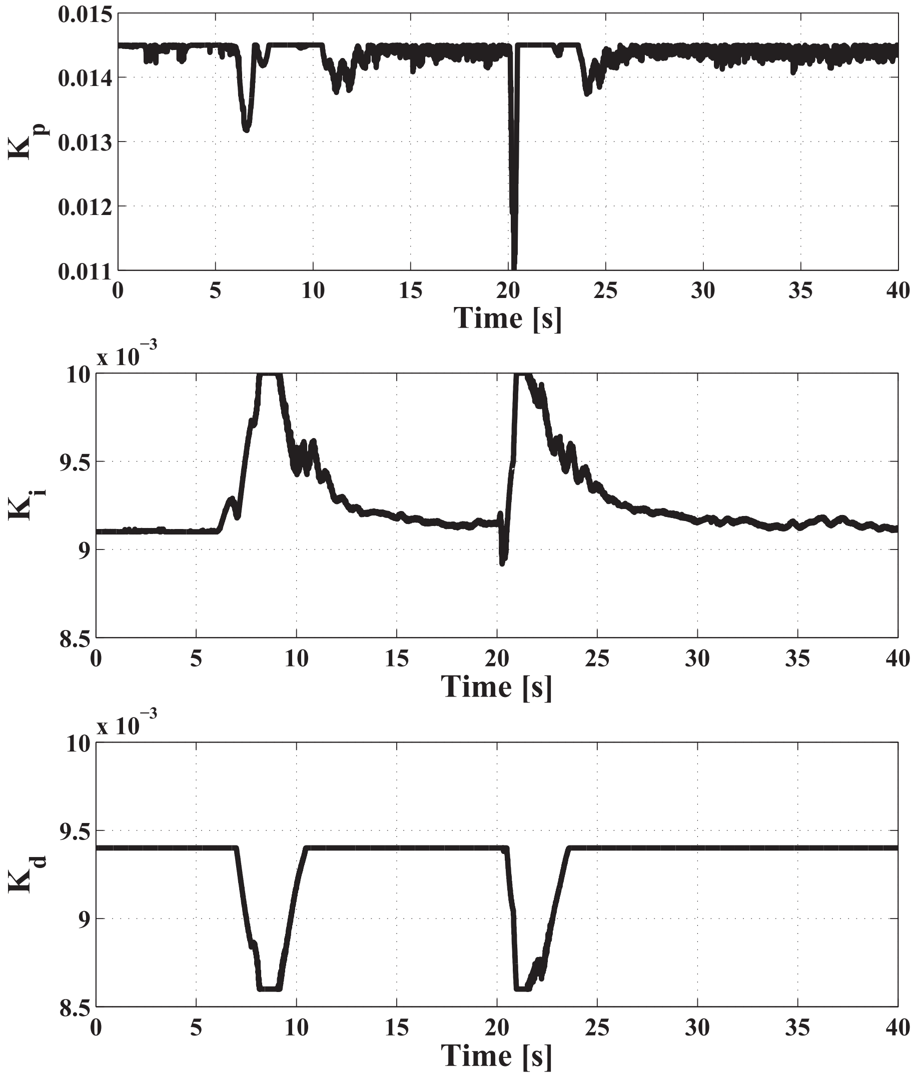


Figure 5.33: Gains K_p, K_i, and K_d in the first scenario.

back to its desired position. After the fault, integrator gain K_i also increased to help the recovery process.

Table 5.7 gives a quantitative comparison between the conventional and the fuzzy PID. The Root Mean Square (RMS) is calculated for the tracking error before fault occurrence and for the 5 seconds after fault. One can see that before fault occurrence, the performance of both controllers are close. However, in the fault case the fuzzy PID greatly reduces tracking error.

Table 5.7: RMS of tracking error

	Before Fault		**After Fault**	
	Conv.	**Fuzzy**	**Conv.**	**Fuzzy**
z-direction	88×10^{-4}	84×10^{-4}	127×10^{-4}	98×10^{-4}

5.5.2 Second Fault Scenario

In the second fault scenario, it is assumed that a loss of control effectiveness of 20% is taking place in the third motor. This kind of fault results in a loss of altitude and drift along the y direction. The gains and predefined ranges for the PID controllers along the z-direction remain the same as given in the previous section. The gains of the conventional PID for the y-direction are $K_p = 0.2137$, $K_i = 0.258$, and $K_d = 0.238$. The predefined ranges of K_p, K_i, and K_d for the fuzzy PID in the y-direction are $K_{p,\min} = 0.09$, $K_{p,\max} = 0.35$, $K_{i,\min} = 0.13$, $K_{i,\max} = 0.35$, $K_{d,\min} = 0.023$, and $K_{d,\max} = 0.029$. Figure 5.34 shows a comparison between the conventional and the fuzzy PID controllers along the height and y-direction. As in the first scenario, the fuzzy PID allows the system to react and return faster to its hovering position. The time evolutions of the fuzzy PID gains are illustrated in Figure 5.35. These gains are related to the fuzzy PID controller in the y-direction.

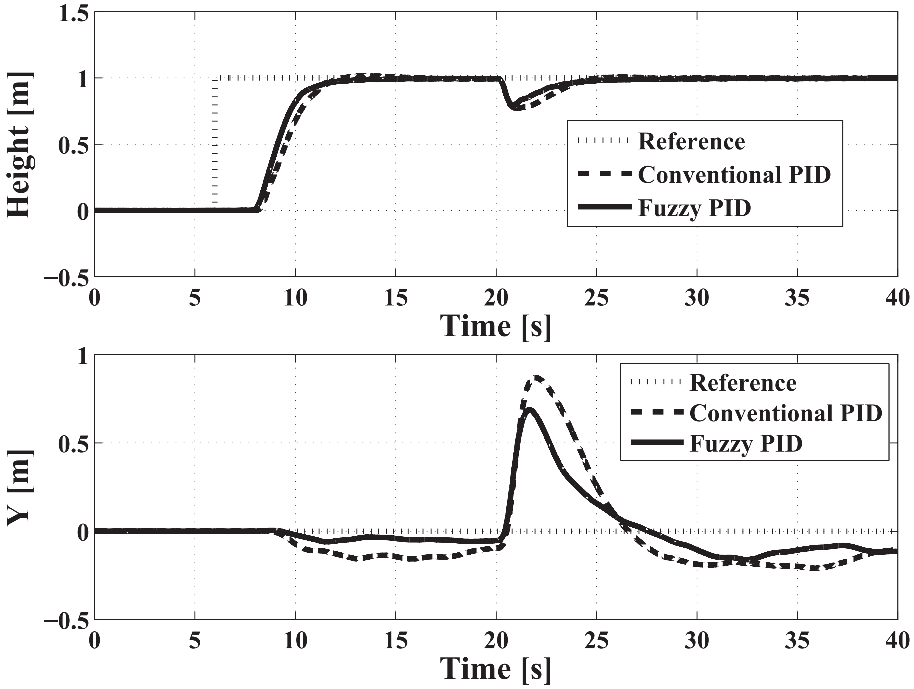


Figure 5.34: Comparison between conventional and fuzzy PID.

As in the first scenario, Table 5.8 gives an quantitative comparison between both controllers using the RMS of the tracking errors along z and y directions. Once again, the fuzzy PID improves system's performance specially when faults occur.

Table 5.8: RMS of tracking errors

	Before Fault		**After Fault**	
	Conv.	**Fuzzy**	**Conv.**	**Fuzzy**
z-direct.	89×10^{-4}	84×10^{-4}	44×10^{-4}	32×10^{-4}
y-direct.	21×10^{-4}	7.6×10^{-4}	191×10^{-4}	132×10^{-4}

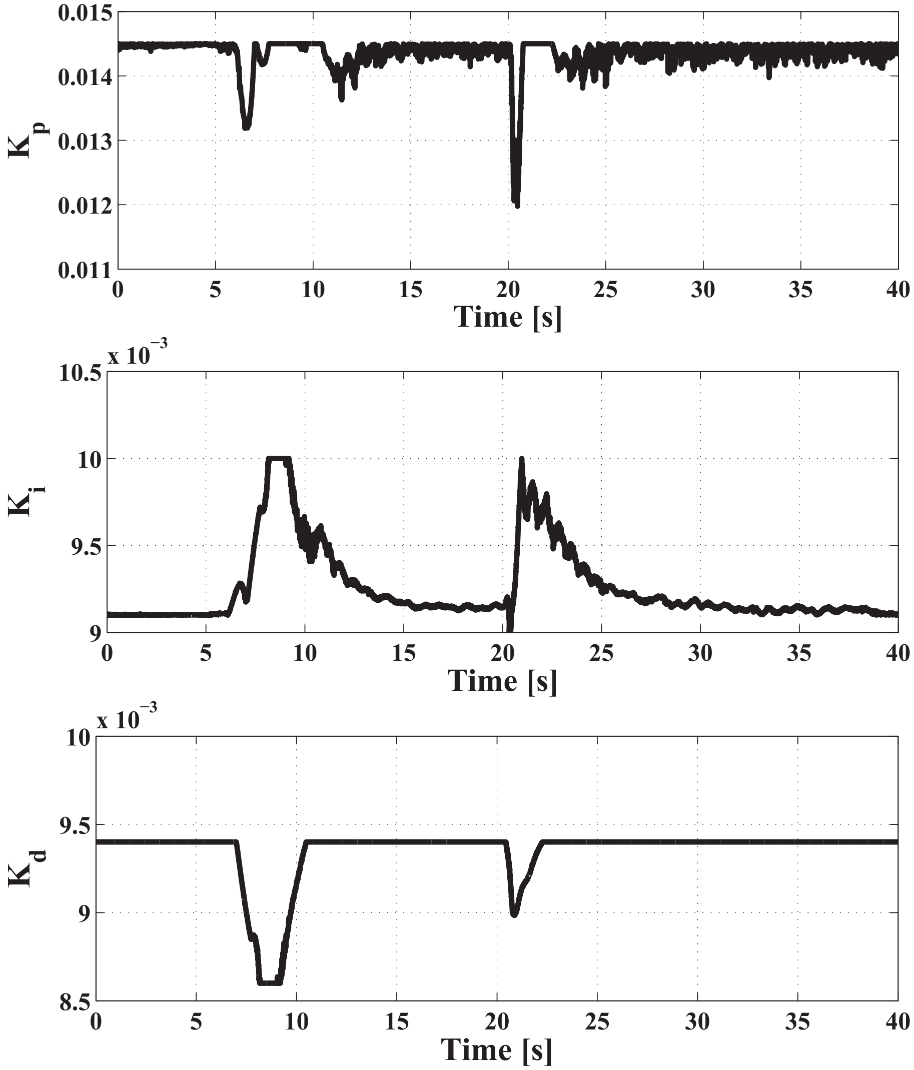


Figure 5.35: Gains K_p, K_i, and K_d in the second scenario.

5.6 Chapter Summary

In this chapter the simulation and experimental results of applying developed FDD methods in Chapter 3 and Controllers developed in Chapter 4 into testbeds described in Chapter 2 have been presented and discussed.

Chapter 6

Conclusions and Future Work

In this chapter the main contributions of this thesis are summarized. Further, potential extensions of the developed methods are then discussed.

6.1 Conclusions

The main focus of this thesis is to develop trajectory tracking and formation controllers under normal (fault-free) conditions, Fault Detection and Diagnosis (FDD) algorithms, and Fault Tolerant Controllers (FTC), to enhance performance, safety and reliability of various missions using unmanned systems. In the following, the limitations raised in Chapter 1 are revisited considering the contributions of this work.

- A new kinematical control method has been proposed for the trajectory tracking of nonholonomic WMRs. The controller is designed based on the concept of guiding the robot to its proper orientation in each instant. In order to derive an appropriate heading angle for the robot, a heading angle scheduler was devised. To investigate the effectiveness of the proposed approach, it has been compared with three well-known algorithms: Model Predictive Control (MPC), Linear State Tracking Control (LSTC) and Nonlinear State Tracking Control (NSTC) methods in experimental tests based on a three-wheeled mobile robot, called Qbot. The real world implementation confirms the advantages with the simplicity, and improved tracking performance with less and smoother control efforts requirements in the proposed controller.
- A stable leader-follower formation controller for time-varying formation configuration of multiple nonholonomic wheeled mobile robots has been presented. It has

been assumed the transition between two configurations is described as a function of time in both separation distance and bearing angle. Then a stable controller was designed to keep the formation during the course of switching between two configurations. The obtained simulation result shows that the group of robots can maintain their desired configuration even during the transition period.

- Two-Stage Kalman Filter (TSKF), Adaptive Two-Stage Kalman Filter (ATSKF), and Interacting Multiple Model (IMM) filter were proposed for Fault Detection and Diagnosis (FDD) of the quadrotor helicopter testbed in the presence of actuator faults. The developed FDD algorithm were evaluated through experimental application to a quadrotor helicopter testbed called *Qball-X4*. Two fault scenarios were investigated: the loss of control effectiveness in all actuators and the loss of control effectiveness in one single actuator. The obtained results showed the effectiveness of the proposed method in terms of both rapid fault detection, correct isolation, and accurate identification of faults.

- FDD algorithm for the attitude control system of the Japan Canada Joint Collaboration Satellite - Formation Flight (JC2Sat-FF) mission has been developed. The FDD scheme was achieved using an Interacting Multiple Model (IMM)-based FDD algorithm. The efficiency of the FDD algorithm has been shown through simulation results in a nonlinear simulator of the JC2Sat-FF. Two different faults in viscous friction and current coefficient have been simulated. From the simulation results it is observed that the algorithm performs satisfactorily in terms of fault detection and diagnosis objectives.

- A fuzzy gain-scheduled PID controller has been designed for a quadrotor unmanned helicopter in the presence of actuator faults. The proposed controller has been tested and compared with the conventional PID controller based on experimental test of the Qball-X4 UAV. The obtained results revealed the effectiveness of the proposed method and its ability to adapt in the presence of uncertainties and external disturbances.

6.2 Future Work

Here after some suggestions and comments regarding further extension of this work is presented.

- For trajectory tracking control of wheeled mobile robots, perfect velocity tracking is assumed and a kinematical controller is proposed. One may design a controller based on the same strategy and in dynamic level.

- The suggested formation controller were examined through simulation, while implementation on a real set of mobile robots can make a huge impact on the validity of it.

- Some FDD algorithms have been successfully implemented on a quadrotor test-bed. The obtained results were promising. As an extension, one may design reconfigurable fault tolerant model-based controller(s) to form a truly active fault tolerant control system.

Bibliography

[1] J. Lee, A. DeRuiter, A. Ng, C. Lambert, Y. Kim, and K. Yhoshihara, “Attitude determination and control subsystem of JC2Sat-FF mission,” in *Proceedings of the 12th International Space Conference of Pacific-basin Societies (ISCOPS)*, Montreal, Canada, July 2010.

[2] D. Chwa, “Tracking control of differential-drive wheeled mobile robots using a backstepping-like feedback linearization,” *IEEE Trans. on Systems, Man, and Cybernetic, Part A: Systems And Humans*, vol. 40, no. 6, pp. 1285–1295, 2010.

[3] M. H. Amoozgar, S. H. Sadati, and K. Alipour, “Trajectory tracking of wheeled mobile robots using a kinematical fuzzy controller,” *Int. J. of Robotics and Automation*, vol. 27, no. 1, pp. 49–56, 2012.

[4] F. Sharifi, M. Mirzaei, B. Gordon, and Y. M. Zhang, “Fault-tolerant control of a quadrotor UAV using sliding mode control,” in *International Conference on Control and Fault-Tolerant Systems*, Nice, France, 6-8 October 2009, pp. 239–244.

[5] Y. M. Zhang and J. Jiang, “Active fault-tolerant control system against partial actuator failures,” *IEE Proceedings on Control Theory Applications*, vol. 149, no. 1, pp. 95–104, 2002.

[6] S. Morgan, “Fault protection techniques in JPL spacecraft,” in *First International Forum on Integrated System Health Engineering and Management in Aerospace*, Napa, California, 2005.

[7] X. Olive, “FDI(R) for satellite at Thales Alenia space: How to deal with high availability and robustness in space domain,” in *International Conference on Control and Fault Tolerant Systems*, Nice, France, October 2010.

[8] J. G. Ziegler and N. B. Nichols, “Optimum settings for automatic controllers,” *Transaction of the ASME*, no. 64, pp. 759–768, 1942.

[9] H. Chen, M. Ma, H. Wang, Z. Liu, and Z. Cai, “Moving horizon H_∞ tracking control of wheeled mobile robots with actuator saturation,” *IEEE Trans. on Control System Technology*, vol. 17, no. 2, pp. 449–457, 2009.

[10] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, “A stable tracking control method for a non-holonomic mobile robot,” in *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, Osaka, Japan, 1991, pp. 1236–1241.

[11] S. H. Sadati, K. Alipour, and M. Behroozi, “A combination of neural network and ritz method for robust motion planning of differentially-driven mobile robots along calculated modular paths,” *Int. J. of Robotics and Automation*, vol. 23, no. 3, pp. 187–198, 2008.

[12] I. Kolmanovsky and N. H. McClamroch, “Developments in nonholonomic control problems,” *IEEE Control Systems Magazine*, vol. 15, pp. 20–36, 1995.

[13] B. Tsiotras and P. Kim, “Controllers for unicycle-type wheeled robots: New theoretical results and experimental validation,” *IEEE Trans. on Robotics and Automation*, vol. 18, no. 3, p. 294307, 2002.

[14] G. Oriolo, A. Luca, and M. Vandittelli, “WMR control via dynamic feedback linearization: Design, implementation, and experimental validation,” *IEEE Trans. on Control Systems Technology*, vol. 10, p. 835852, 2002.

[15] F. Pourboghrat and M. P. Karlsson, “Adaptive control of dynamic mobile robots with nonholonomic constraints,” *Int. J. of Computers and Electrical Engineering*, vol. 28, p. 241253, 2002.

[16] G. Klancar and I. Skrjanc, “Tracking-error model-based predictive control for mobile robots in real time,” *Int. J. of Robotics and Autonomous Systems*, vol. 55, pp. 460–469, 2007.

[17] K. Kanjanawanishkul, M. Hofmeister, and A. Zell, “Smooth reference tracking of a mobile robot using nonlinear model predictive control,” in *European Conf. on Mobile Robots (ECMR)*, Mlini/Dubrovnik, Croatia, 2009.

[18] S. J. Yoo, “Adaptive tracking control for a class of wheeled mobile robots with unknown skidding and slipping,” *IET Control Theory and Applications*, vol. 4, no. 10, pp. 2109–2119, 2010.

[19] Z. Cao, Y. Zhao, and Q. Wu, “Adaptive trajectory tracking control for a nonholonomic mobile robot,” *Chinese J. of Mechanical Engineering*, vol. 24, no. 3, pp. 1–7, 2011.

[20] R. Fierro and F. L. Lewis, “Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics,” *Int. J. of Robotic System*, vol. 14, no. 3, pp. 149–163, 1997.

[21] Y. Duan, B. Cui, and H. Yang, “Robot navigation based on Fuzzy RL algorithm,” *Springer-Verlag Berlin Heidelberg*, pp. 391–399, 2008.

[22] P. Morin and C. Samson, “Control of nonholonomic mobile robots based on the transverse function approach,” *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1058–1073, 2009.

[23] K. G. Jolly, R. S. Kumar, and R. Vijayakumar, “An artificial neural network based dynamic controller for a robot in a multi-agent system,” *Int. J. of Neurocomputing*, vol. 73, pp. 283–294, 2009.

[24] Z. Hendzel and M. Szuster, “Discrete neural dynamic programming in wheeled mobile robot control,” *J. of Communication, Nonlinear Science and Numerical Simulation*, vol. 16, no. 5, pp. 2355–2362, 2010.

[25] C. Chen, S. Li, Y. Yeh, and C. Chang, “Design and implementation of an adaptive sliding-mode dynamic controller for wheeled mobile robots,” *Int. J. of Mechatronics*, no. 19, pp. 156–166, 2009.

[26] M. K. Bugeja, S. G. Fabri, and L. Camilleri, “Dual adaptive dynamic control of mobile robots using neural networks,” *IEEE Trans. on Systems, Man, And Cybernetics, Part B: Cybernetics*, vol. 39, no. 1, pp. 129–141, 2009.

[27] S. D. Wang and C. K. Lin, “Adaptive tuning of the fuzzy controller for robots,” *Int. J. of Fuzzy Sets and Systems*, vol. 110, no. 2, pp. 351–363, 2000.

[28] R. Martinez, O. Castillo, and L. Aguilar, “Intelligent control for a perturbed autonomous wheeled mobile robot using type-2 fuzzy logic and genetic algorithms,” *Int. J. of Automation, Mobile Robotics and Intelligent Systems*, no. 2, pp. 12–22, 2008.

[29] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and J. Taylor, "A vision-based formation control framework," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 5, pp. 813–825, 2002.

[30] M. Sisto and D. Gu, "A fuzzy leader-follower approach to formation control of multiple mobile robots," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 2515–2520.

[31] M. Defoort, T. Floquet, A. Kksy, and W. Perruquett, "Sliding-mode formation control for cooperative autonomous mobile robots," *IEEE Trans. on Industrial Electronics*, vol. 55, no. 11, pp. 3944–3953, 2008.

[32] R. M. Bhatt, C. P. Tang, and V. N. Krovi, "Formation optimization for a fleet of wheeled mobile robots: A geometric approach," *Journal of Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 102–120, 2009.

[33] R. R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 34, no. 2, pp. 138–153, 2004.

[34] I. R. Nourbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion, "Human-robot teaming for search and rescue," *IEEE Pervasive Computing*, vol. 4, no. 2, pp. 72–78, 2005.

[35] P. Varaiya, "Smart cars on smart roads: Problems of control," *IEEE Trans. on Automatic Control*, vol. 38, no. 2, pp. 195–207, February 1993.

[36] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multi-agent hybrid systems," *IEEE Trans. on Automatic Control*, vol. 34, no. 4, pp. 509–521, May 1993.

[37] D. Voth, "A new generation of military robots," *IEEE Intelligent Systems*, vol. 19, no. 4, pp. 2–3, 2004.

[38] W. Ren and R. Beard, "A decentralized scheme for spacecraft formation flying via the virtual structure approach," in *Proc. American Control Conf.*, Denver, USA, 2003, pp. 1746–1751.

[39] K. Tan and M. A. Lewis, "Virtual structures for high precision cooperative control," in *Technical Report*, Computer Science Department, University of California, Los Angles, 1997.

[40] W. Ren and R. W. Beard, "A decentralized scheme for spacecraft formation flying via the virtual structure approach," *AIAA Journal of Guidance, Control and Dynamics*, vol. 27, no. 1, pp. 73–78, 2004.

[41] M. Egerstedt and K. Hu, "Formation constrained multi-agent control," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 6, pp. 947–951, December 2001.

[42] J. Ghommama, H. Mehrjerdi, M.Saad, and F. Mnif, "Formation path following control of unicycle-type mobile robots," *Journal of Robotics and Autonomous Systems*, vol. 58, pp. 727–736, 2010.

[43] A. Bazoula, M. Djouadi, and H. Maaref, "Formation control of multi-robots via fuzzy logic technique," *Journal of Computers, Communications and Control*, vol. 3, pp. 179–184, 2008.

[44] J. Esposito and V. Kumar, "A formalism for parallel composition of reactive and deliberative control objectives for mobile robots," in *Technical Report*, Mechanical Eng. and Applied Mechanics, Univ. of Pennsylvania, Philadelphia, 2000.

[45] H. Yamgachi, "A cooperative hunting behavior by mobile robot troops," in *IEEE Int. Conference on Robotics and Automation (ICRA)*, Leuven, Belgium, 1998, pp. 3204–3209.

[46] J. Xiang and T. Braunl, "String formations of multiple vehicles via pursuit strategy," *IET Control Theory and Applications*, vol. 4, pp. 1027–1038, 2010.

[47] J. Sanchez and R. Fierro, "Sliding mode control for robot formations," in *Proc. of IEEE Int. Symposium on Intelligent Control*, Houston, Texas, USA, 2003, pp. 438–443.

[48] X. Li and J. Xiao, "Robot formation control in leaderfollower motion using direct lyapunov method," *Int. Journal of Control System*, vol. 10, pp. 244–250, 2005.

[49] T. Dierks and S. Jagannathan, "Control of nonholonomic mobile robot formations: Backstepping kinematics into dynamics," in *Proc. of the IEEE Multi-Conf. Systems Control*, Singapore, 2007, pp. 94–99.

[50] K. Choi, S. J. Yoo, J. B. Park, and Y. H. Choi, "Adaptive formation control in absence of leaders velocity information," *IET Control Theory and Applications*, vol. 4, pp. 521–528, 2010.

[51] L. Consolini, F. Morbidi, and D. Prattichizzo, “Stabilization of a hierarchical formation of unicycle robots with velocity and curvature constraints,” *IEEE Trans. on Robotics*, vol. 25, no. 5, pp. 1176–1184, 2009.

[52] T. Yang, Z. Liu, H. Chen, and R. Pei, “Formation control and obstacle avoidance for multiple mobile robots,” *Journal of ACTA Automatica Sinika*, vol. 34, no. 5, pp. 588–593, 2008.

[53] S. M. de Oca, V. Puig, D. Theilliol, and S. Tornil-Sin, “Fault-tolerant control design using LPV admissible model matching: Application to a two-degree of freedom helicopter,” in *17th Mediterranean Conference on Control and Automation*, Thessaloniki, Greece, 24-26 June 2009, pp. 522–527.

[54] F. Wen, P. Willett, and S. Deb, “Signal processing and fault detection with application to CH-46 helicopter data,” in *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, USA, March 2000, pp. 15–26.

[55] G. Heredia, A. Ollero, M. Bejar, and R. Mahtani, “Sensor and actuator fault detection in small autonomous helicopter,” *Mechatronics*, vol. 18, no. 2, pp. 90–99, 2008.

[56] G. Heredia, V. RemuB, A. Ollero, R. Mahtani, and M. Musial, “Actuator fault detection in autonomous helicopters,” in *Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, July 2004.

[57] N. Zhang, A. Doncescu, and F. Mora-Camino, “Discrete flatness for non linear systems FDI,” in *International Conference on Broadband, Wireless Computing, Communication and Applications*, Fukuoka, Japan, 4-6 November 2010c, pp. 448–494.

[58] N. Zhang, A. Drouin, A. Doncescu, and F. Mora-Camino, “A differential flatness approach for rotorcraft fault detection,” in *27th Chinese Control Conference*, Kunming, Yunnan, China, 2008b, pp. 237–241.

[59] A. Freddi, S. Longhi, and A. Monteriu, “A model-based fault diagnosis system for a mini-quadrotor,” in *7th Workshop on Advanced Control and Diagnosis*, Zielona Gora, Poland, 19-20 November 2009.

[60] ——, “Actuator fault detection system for a mini-quadrotor,” in *IEEE International Symposium on Industrial Electronics*, Bari, Italy, 4-7 July 2010, pp. 2055–2060.

[61] C. Berbra, S. Lesecq, and J. J. Martinez, "A multi-observer switching strategy for fault-tolerant control of a quadrotor helicopter," in *16th Mediterranean Conference on Control and Automation*, Ajaccio, Corsica, France, 2008, pp. 1094–1099.

[62] H. Rafaralahy, E. Richard, M. Boutayeb, and M. Zasadzinski, "Simultaneous observer based sensor diagnosis and speed estimation of unmanned aerial vehicle," in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008, pp. 2938–2943.

[63] H. V. Nguyen, C. Berbra, S. Lesecq, S. Gentil, A. Barraud, and C. Godin, "Diagnosis of an inertial measurement unit based on set membership estimation," in *The 17th Mediterranean Conference on Control and Automation*, Thessaloniki, Greece, 24-26 June 2009, pp. 211–216.

[64] A. de Ruiter, J. Lee, and A. Ng, "Fault tolerant magnetic spin stabilizing controller for the JC2Sat-FF mission," in *AIAA Guidance, Navigation and Control Conference*, Hawaii, USA, August 2008.

[65] E. S. Tehrani and K. Khorsani, "Fault diagnosis of nonlinear systems using a hybrid approach," *Lecture Notes in Control and Information Sciences*, vol. 383, 2009.

[66] E. S. Tehrani, K. Khorasani, and S. Tafazoli, "Dynamic neural network-based estimator for fault diagnosis in reaction wheel actuator of satellite attitude control system," in *Proceeding of International Joint Conference on Neural Networks*, 2005, pp. 2347–2352.

[67] Y. M. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault tolerant control systems," *Annual Reviews in Control*, vol. 32, no. 2, pp. 229–252, 2008.

[68] J. Y. Keller and M. Darouach, "Optimal two-stage Kalman filter in the presence of random bias," *Automatica*, vol. 33, no. 9, pp. 1745–1748, September 1997.

[69] N. E. Wu, Y. M. Zhang, and K. M. Zhou, "Control effectiveness estimation using an adaptive Kalman estimator," in *Proceedings of IEEE ISIC/CIRA/ISAS Joint Conference*, Gaithersburg, MD, USA, 14-17 September 1998, pp. 181–186.

[70] Y. M. Zhang and N. E. Wu, "Fault diagnosis for a ship propulsion benchmark: Part I," in *Preprints of the 14th IFAC World Congress*, Beijing, China, 5-9 July 1999.

[71] N. E. Wu, Y. M. Zhang, and K. M. Zhou, “Detection, estimation, and accommodation of loss of control effectiveness,” *International Journal of Adaptive Control and Signal Processing. Special Issue: Fault Detection and Isolation*, vol. 14, no. 7, pp. 775–795, November 2000.

[72] Y. M. Zhang and X. R. Li, “Detection and diagnosis of sensor and actuator failures using IMM estimator,” *IEEE Transactions of Aerospace and Electronics Systems*, vol. 34, no. 4, pp. 1293–1313, 1998.

[73] L. Cork and R. Walker, “Sensor fault detection for UAVs using a nonlinear dynamic model and the IMM-UKF algorithm,” in *Information, Decision and Control*, Adelaide, Qld., Australia, 12-14 February 2007, pp. 230–235.

[74] L. Ma and Y. M. Zhang, “DUKF-based fault detection and diagnosis for GTM UAV using nonlinear and LPV models,” in *Proceedings of 2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, Qingdao, P. R. China, July 2010, pp. 275–380.

[75] T. C. T. Ng, F. H. F. Leung, and P. K. S. Tam, “A simple gain scheduled PID controller with stability consideration based on a grid-point concept,” in *Proceedings of the IEEE International Symposium on Industrial Electronics*, Guimaraes, Portugal, 1997, pp. 1090–1094.

[76] F. Karray, W. Gueaieb, and S. Al-Sharhan, “The hierarchical expert tuning of PID controllers using tools of soft computing,” *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 32, no. 1, pp. 77–90, 2002.

[77] Z. Zhao, M. Tomizuka, and S. Isaka, “Fuzzy gain scheduling of PID controllers,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 5, pp. 1392–1398, 1993.

[78] K. Yu and J. Hsu, “Fuzzy gain scheduling PID control design based on particle swarm optimization method,” in *Second International Conference on Innovative Computing, Information and Control*, Kumamoto, 2007.

[79] Zulfatman and M. F. Rahmat, “Application of self-tuning fuzzy PID controller on industrial hydraulic actuator using system identification approach,” *Int. J. on Smart Sensing and Intelligent Systems*, vol. 2, pp. 246–261, 2009.

[80] Y. Guo and T. Yang, “A new type of computational verb gain-scheduling PID controller,” in *International Conference on Counterfeiting Security and Identification in Communication*, Chengdu, 2010, pp. 235–238.

[81] L. Yao and C. Lin, “Design of gain scheduled fuzzy PID controller,” *World Academy of Science, Engineering and Technology*, no. 1, pp. 152–156, 2005.

[82] I. M. M. El Emary, W. Emar, and M. J. Aqel, “The adaptive fuzzy designed PID controller using wavelet network,” *Journal of Computer Science and Information System*, vol. 6, no. 2, pp. 141–163, 2009.

[83] B. Hu, G. K. I. Mann, and R. G. Gosine, “A systematic study of fuzzy PID controllers-function-based evaluation approach,” *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 5, pp. 699–712, 2001.

[84] I. Sadeghzadeh, A. Mehta, Y. M. Zhang, and C. A. Rabbath, “Fault-tolerant trajectory tracking control of a quadrotor helicopter using gain-scheduled PID and model reference adaptive control,” in *Annual Conference of the PHM Society*, Montreal, QC, Canada, 25-29 September 2011.

[85] K. Yoshihara, M. van Mierlo, A. Ng, B. S. Kumar, A. D. Ruiter, Y. Komatsu, H. Horiguchi, and H. Hashimoto, “JC2Sat-FF: An international collaboration nanosat project overview of the project, system analyses and design,” in *Proceedings of the IAA Symposium on Small Satellite Systems and Services (4S)*, Rhodes, Greece, May 2008.

[86] T. Dierks and S. Jagannathan, “Neural network output feedback control of a quadrotor UAV,” in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, 9-11 December 2008, pp. 3633–3639.

[87] ——, “Neural network control of quadrotor UAV formations,” in *American Control Conf.*, St. Louis, Missouri, USA, 10-12 June 2009, pp. 2990–2996.

[88] SWARM, http://vertol.mit.edu, 2011, [cited 30 October 2011].

[89] STARMAC, http://hybrid.stanford.edu/~starmac/project.htm, 2011, [cited 30 October 2011].

[90] MAST, https://www.grasp.upenn.edu/research/micro_autonomous_system_technologies_mast, 2011, [cited 30 October 2011].

[91] NAV, http://users.encs.concordia.ca/~ymzhang/UAVs.htm, 2011, [cited 30 October 2011].

[92] Quanser, “Quanser Qball-X4 User Manual,” 2010, document number 829.

[93] T. Bresciani, “Modelling, Identification and Control of a Quadrotor Helicopter,” Master’s thesis, Lund University, Sweden, 2008.

[94] J. C. Alexander and J. H. Maddocks, “On the kinematics of wheeled mobile robots,” *Int. J. of Robotics Research*, vol. 8, no. 5, pp. 15–27, 1989.

[95] D. Sinclair, C. C. Grant, and R. E. Zee, “Enabling reaction wheel technology for high performance nanosatellite attitude control,” in *21st Annual AIAA/USU Conference on Small Satellites*, Logan, UT, 2007.

[96] J. E. Parkum, N. K. Poulsen, and J. Holst, “Recursive forgetting algorithms,” *International Journal of Control*, vol. 55, pp. 109–128, 1992.

[97] H. K. Khalil, *Nonlinear Systems*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.

[98] S. X. Ding, *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Springer, 2008.

[99] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer, 2006.

[100] P. Puschner and C. Koza, “Calculating the maximum execution time of real-time programs,” *Real-Time Systems*, vol. 1, pp. 159–176, 1989, 10.1007/BF00571421. [Online]. Available: http://dx.doi.org/10.1007/BF00571421