2022

Fault-Tolerant Control of Autonomous Ground Vehicle under Actuator and Sensor

Vaishnavi Janakiraman
Wright State University

FAULT-TOLERANT CONTROL OF AUTONOMOUS GROUND VEHICLES UNDER ACTUATOR AND SENSOR FAULTS

A thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Electrical Engineering

By

VAISHNAVI JANAKIRAMAN

B.E., Anna University, India, 2018

2022

Wright State University

WRIGHT STATE UNIVERSITY

GRADUATE SCHOOL

December 6, 2022

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY **Vaishnavi Janakiraman** ENTITLED **Fault-Tolerant Control of Autonomous Ground Vehicles under Actuator and Sensor Faults** BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF **Master of Science in Electrical Engineering**

_____

Xiaodong Zhang, Ph.D.

Thesis Director

_____

Michael A. Saville, Ph.D.

Associate Professor and Chair,

Department of Engineering & Computer Science

Committee on Final Examination:

_____

Xiaodong Zhang, Ph.D.

_____

Luther Palmer III, Ph.D.

_____

Yan Zhuang, Ph.D.

_____

Shu Schiller, Ph.D.

Interim Dean of the Graduate School

ABSTRACT

Janakiraman, Vaishnavi, M.S.E.E., Department of Electrical Engineering, Wright State University, 2022. Fault-Tolerant Control of Autonomous Ground Vehicles under Actuator and Sensor Faults

Unmanned ground vehicles have a wide range of potential applications including autonomous driving, military surveillance, emergency responses, and agricultural robotics, etc. Since such autonomous vehicles need to operate reliably at all times, despite the possible occurrence of faulty behaviors in some system components, the development of fault-tolerant control schemes is a crucial step in ensuring reliable and safe operations. In this research, a fault-tolerant control scheme is developed for a nonlinear ground vehicle model with possible occurrence of both actuator faults in the form of loss of effectiveness (LOE) and sensor bias faults. Based on the vehicle and fault models under consideration, the unknown fault parameters are estimated online using adaptive estimation methods. The estimated fault parameters are used for accommodating the fault effect to maintain satisfactory control performance even in the presence of faults. Real-time algorithm implementation and demonstration using the Qbot2e ground robot by Quanser are conducted to show the effectiveness of the fault-tolerant control algorithm.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to take this opportunity to thank my advisor. Dr Xiaodong Zhang (Frank) for his endless support and guidance throughout my master's degree along with his sincere care for my future career path. I have learnt a lot working with him and have been highly influenced by his in-depth knowledge in the field of robotics and control system. I feel truly fortunate to have him as my advisor and have the opportunity to work with him.

I would also like to thank my parents for being my strength at all times, loving and motivating me at every hurdle of my life. Not to forget I would like to thank my fellow friend for his useful discussions enlightening me with great ideas to tackle the problem seeing through a different angle.

Last but not the least, I would specially thank my husband for his optimism, constant encouragement, and never-ending motivation to get through these tough times with at most ease.

1 INTRODUCTION

1.1 Fault Diagnosis, Estimation and Accommodation (An Overview)

Faults are the undesirable changes in the normal functioning behavior of the robot. Faults can potentially degrade the robot's performance, or even lead to unstable or unsafe behaviors. Faults can be categorized as hardware and software faults. Software faults are bugs in the control algorithm and software code. As described in [1,5], hardware faults can be categorized into three subcategories:

- Sensor faults corrupt the system measurements, causing a deviation between actual and sensed input/output variables,
- Actuator faults represent discrepancies between commanded input and actual input to the system,
- Process/component faults are physical failures of the system components which lead to changes in the normal system dynamics.

In this research, we focus on actuator and sensor faults in ground robotic systems. These faults can cause degraded path-following capabilities of the robot, or even catastrophic effects. To ensure safe and reliable operations of the robotic systems, fault-tolerant control algorithms must be developed to accommodate the undesirable effects of the faults, so that acceptable control performance is maintained even in the presence of such faults.

To detect and accommodate several types of faults in the system, the fault-tolerant control procedure is further subdivided into three main steps:

- *Fault Detection and Isolation* determines the occurrence of the faults in the system components and localize the specific faulty components.
- *Fault Estimation* provides an estimate of the fault size. Fault estimation helps in determining the nature of the fault and thus facilitates the design of the accommodation strategies.
- *Fault Accommodation* corrects the fault effects using the fault information obtained from the above two steps, hence recovering the tracking performance of the control system.

1.2 Literature Survey

Unmanned ground vehicles are autonomous robots that operate automatically without the human intervention. It has various applications including military surveillance, emergency rescue, manufacturing, and areas where human intervention should be minimized. They can conduct operations by observing the environment and taking actions to adapt to environmental changes. Important components of the control system for ground vehicles include path planning, obstacle avoidance, trajectory following, fault detection and fault accommodation, etc.

Faults are undesired changes in the ground robotic system components which are hard to avoid given the hostile environment conditions. Depending on the condition of the fault, various methods have been proposed for fault detection and accommodation of ground robots, which can be sub divided into three main categories:

1. *Data-driven* methods which provide feature extraction and data classification for detecting the faulty behaviors. It includes neural network-based model or

reinforcement learning algorithms. In [7], fault-tolerant control design of wheeled planetary rover was presented. A nominal mode controller is used for trajectory tracking, and the impact of faults due to failure in the steering joints in the system was analyzed. The algorithm presented in [11] identifies faults in wheeled mobile robots by using multiple parallel Kalman Filters as estimators and comparing with the backpropagation neural network for fault detection and accommodation. Kalman filters are used for parameter estimation, and the neural network model is trained to process the residuals to determine which type of mechanical and sensor faults have occurred.

2. *Model-based* methods which uses a specific system model for fault detection and accommodation. In [6], the researcher has presented the use of Multiple Model Adaptive Estimation (MMAE) for identifying different types of faults and a particle filter-based approach for fault estimation. A backstepping sliding mode controller is proposed to improve the performance of the inductive machine in the presence of faults in [12]. Parity space and observer-based fault diagnosis approaches with applications to ground vehicle are presented in [14]. This model-based method is extremely specific to the model making it exceedingly difficult to process with a generic type of application limiting it to a specific application.
3. *Knowledge based* fault detection and diagnosis (FDD) methods [15-18], which are a hybrid of the first two categories involve helping a trade-off between the two approaches in providing best solution for FDD. The hybrid approach for FDD provides solution to overcome lack of data issues and thus increase the FDD accuracy. The design of active fault-tolerant control methods for electrical vehicles is presented in [15]. It uses a baseline controller, a FDD mechanism to accommodate the fault

occurred due to actuator and decision mechanism to achieve better performance after the fault is detected and accommodated. Similar method also called as fault-tolerant cooperative control for wheeled mobile robots are used to estimate the fault presence in an autonomous mobile robot to enhance the fault-tolerance performance of the system using FDD mechanism using both the data-driven and model- based methods for its operation [16]. Triple-step approach-based fault tolerant control (FTC) in [17], does dual purpose of first finding the uncertainties in the mobile vehicles to retain the stability followed by providing high-safety for the closed-loop driver system using model predictive control method and the data-driven method for sample collection. Detailed study on FDD based learning is discussed in [18], which focuses on different attributes of Multi-Robot System (MRS) along with various FDD approaches providing a detailed insight to the FDD for MRS. The 4WDW- EV system model is used along with the data collection at every sample period for fault estimation. Fault Detection and Diagnosis (FDD) based mechanism as described in the above papers, has many challenges to distinguish the fault types in the system due to the degraded performance and diagnosing the root cause of the faults.

1.3 Research Objectives and Contributions

The papers discussed above presented very interesting and powerful fault diagnosis and fault-tolerant control methods for ground robots. However, there exist limited results that consider simultaneous actuator and sensor faults in ground vehicles. For instance, only bias actuator faults were considered in recent work [22] with an assumption that the bias faults in the left and right wheels are of the same value. Additionally, fault-tolerant control

methods that explicitly integrates the tasks of the fault diagnosis and fault accommodation are still lacking.

The objective of this research is to address some of the limitations in the existing research works. Firstly, it considers simultaneous actuator faults and sensor faults in the ground vehicle. Also, the actuator faults in two wheels are allowed to be of different sizes making it more general and suitable for real-time situations. Secondly, fault diagnosis and accommodation tasks are seamlessly integrated by using adaptive parameter estimation techniques. The algorithms are implemented using a real-time testing environment to demonstrate the effectiveness of the fault-tolerant system.

1.4 Thesis Organization

This research is documented in the following order:

- **Chapter 1:** This chapter gives an introduction to the Fault Diagnosis, Estimation and Accommodation. It also involves a literature survey and research objectives of this thesis.
- **Chapter 2:** This chapter briefly explains the hardware and software parts which was used during the process of research. It also describes the robotic system as well as the Vicon camera system used for the thesis.
- **Chapter 3:** This presents the problem statement for the research. It provides an insight to the thesis scope before deep diving to the accomplishments.
- **Chapter 4:** This chapter presents the adaptive fault parameter estimation algorithm for different types of faults in the ground vehicle. It describes the implementation

of the fault-tolerant and fault diagnosis algorithm while dealing with all the faults encountered in the system.

- **Chapter 5:** This chapter shows the real-time experimental results validating the effectiveness of the fault-tolerant control algorithm.
- **Chapter 6:** This last chapter provides some conclusions and remarks about possible directions for future research.

2 HARDWARE AND SOFTWARE

2.1 QBot2e Ground Robot Hardware

This research involves the usage of a ground robot by Quanser also referred to as QBot2e. It is a nonholonomic wheeled mobile robot incorporating ground vehicle with Microsoft Kinect sensor along with Quanser embedded target [3]. QBot2e is interfaced with the MATLAB/Simulink with the QUARC software on the target machine. The Hardware-In-Loop (HIL) block set is the block which handles reading the sensors and writing to the outputs. The RGB and depth image data are collected from the Kinect sensor (not used in this research). The controllers are developed using the Simulink model with QUARC on the host computer, and their models are downloaded to the target which is the Unmanned Ground Vehicle (QBot2e) for it to perform the desired actions as specified by the Simulink model.

This communication between the host computer and the target is shown diagrammatically as follows:

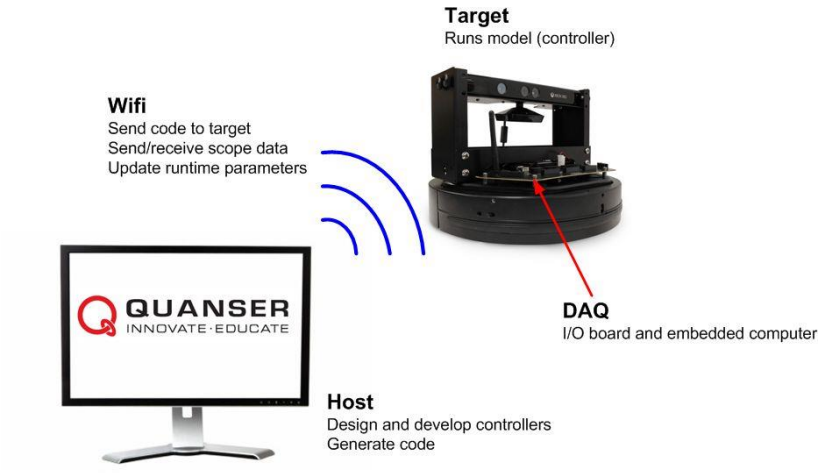


Figure 1: Communication between host computer and the target

The Unmanned Ground Robot has various parts on it which conduct different operations for the robot to follow the desired objective and extract maximum information out of it for better functioning and robust capabilities. Hardware components included in the robot are listed in the following table.

QBot2e Hardware and Software Components	**Description**
Two Central Drive wheels (Right and left drive wheels)	The two wheels are independently driven forward or backward for easy maneuverability.
Castors (Front and Back)	It is used to stabilize and alignment adjustments making the vehicle more stable and predictable under various terrain conditions.
Microsoft X-box Kinect Vision Sensor	Helps extract the depth information from capturing the motion of the robot and is also using Robot Operating System (ROS) providing indoor navigation feature.
Integrated Bumpers (Right, left and front)	It is used to absorb the vibrations caused by the motors preventing damage due to obstacle hitting to the other components in the robotic applications.

Cliff Sensors	Mechanical cliff sensors are used to alert the controller of the robot if it has reached the drop off when it runs on the ground.
HIL Initialize Block	Configures drives and hardware interface with the QBot2e.
HIL Read and Write Block	It used the read sensory data to drive the motors of the robot.
Video 3D capture	It captures the depth and RGB data which can be used for SLAM
Video Compressed Display	It helps in transmitting the RGBD data extracted to the PC from QBot2e for further working with the data collected.

2.2 Vicon

The Vicon camera system is expertized to provide a tailored motion capture system for capturing the motion of the ground robot. It has various applications in fields like life science, media and entertainment, virtual reality based on location, and engineering education.

This platform provides high accuracy positioning system to capture the optical data of the robot when in motion. These four cameras emit infrared light from LED array and the positioning of the object is specified using markers set during the Vicon calibration process. Vicon is also used in the quadrotor system to calculate attitude estimates for roll

and pitch angles. For the ground robot used in this research, the Vicon system provides measurements for the pose of the vehicle, including positions in the X-Y coordinate and the heading angle.

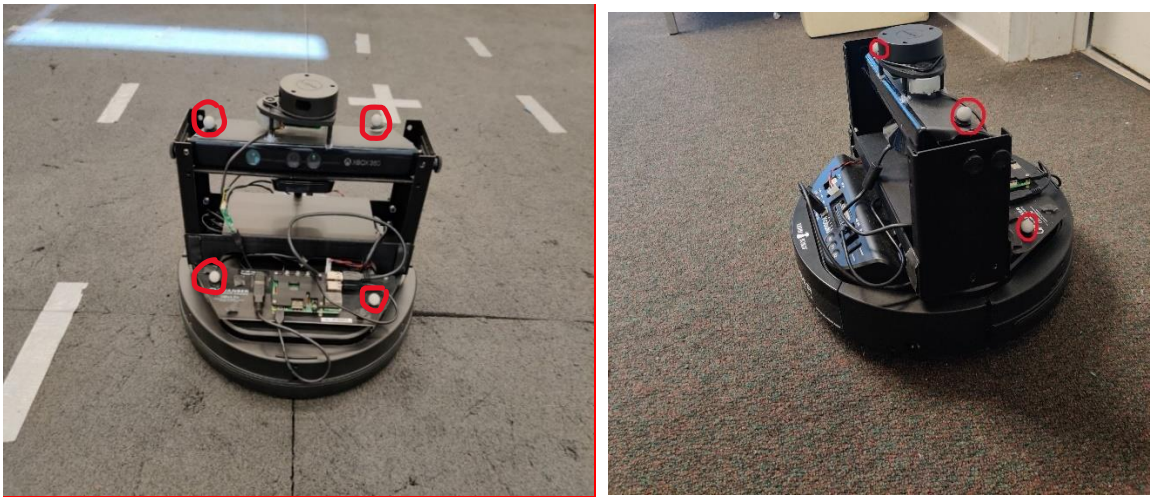


Figure 2: Quanser QBot2e Ground robot with Xbox Kinect and LIDAR from front (left) and side (right)

The ground station or the PC is made to communicate with the ground vehicle and the Vicon system using a wireless network. These cameras are designed to capture the position and altitude of the ground robot. The process starts with creating an object inside the Vicon Tracker 2.0 software so that the environment identifies the robot, and this makes it easy for differentiating between the objects inside the boundary (see Figure 3).

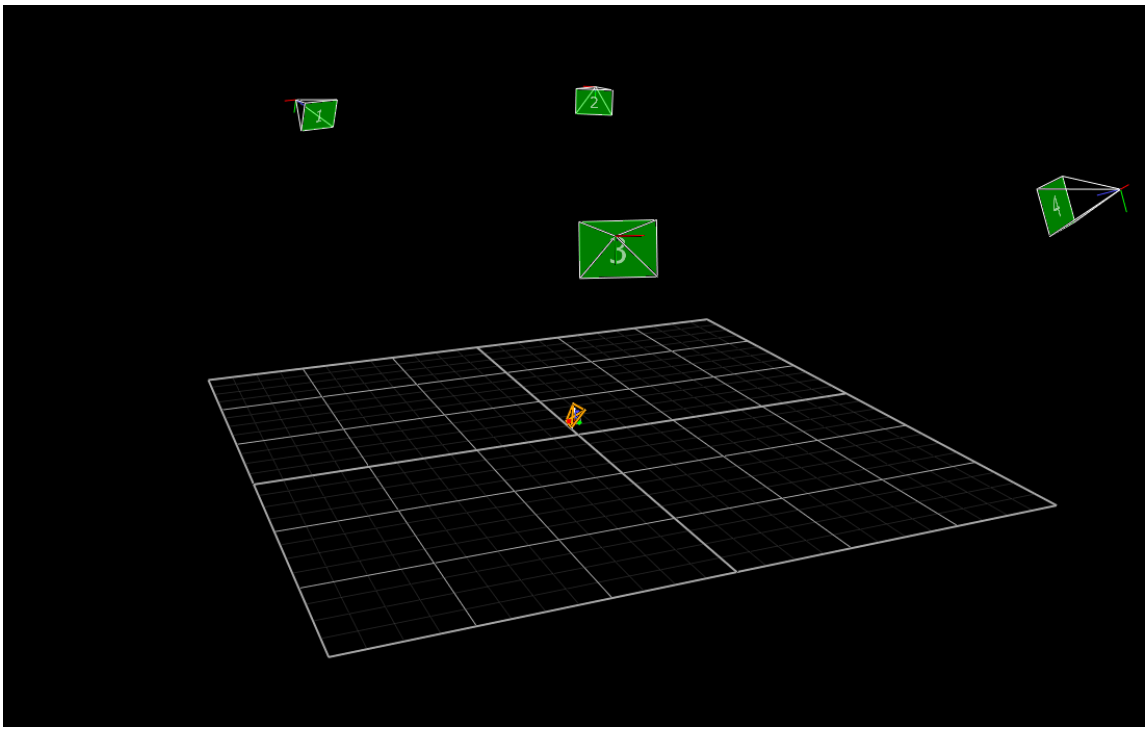


Figure 3: Vicon camera system visualization through Vicon tracker 2.0

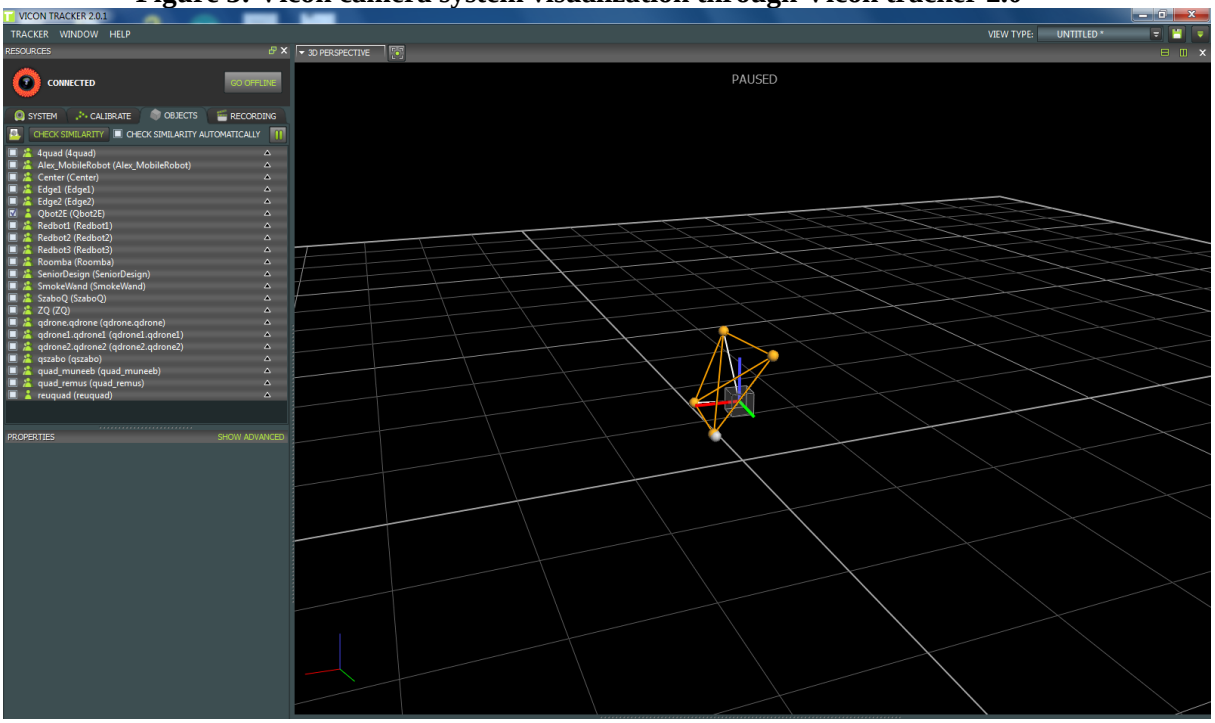


Figure 4: Vicon Tracker 2.0 showing Ground robot using Vicon cameras.

The blocks inside the Simulink model brings in the data collected by the camera and extracts x, y and yaw angles which is then compared with the reference signals coming from the trajectory generation block for tracking purposes.

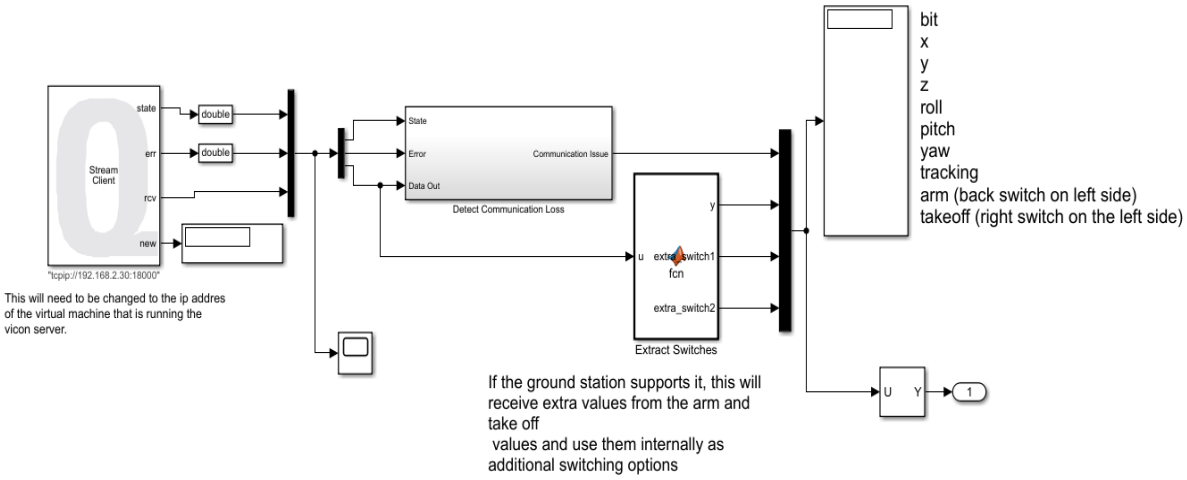


Figure 5: Simulink model showing Vicon system connected to MATLAB to extract poses of the robot.

3 PROBLEM FORMULATION

Let us consider the equation of kinematic model represented by the following state space model:

$$\dot{x} = \frac{(V_r{}^a + V_l{}^a)\, cos\varphi}{2}$$

$$\dot{y} = \frac{(V_r{}^a + V_l{}^a)\, sin\varphi}{2} \quad (1)$$

$$\dot{\varphi} = \frac{(V_r{}^a + V_l{}^a)}{d}$$

where x, y and φ are the position and the heading angle of the robot, $V_r{}^a$ and $V_l{}^a$ are the velocities of the robot right and left wheels, and d is the distance between left and right wheels.

In this research, we consider loss of effectiveness actuator faults and sensor bias faults in the ground vehicle.

Loss of effectiveness in the actuator results in partial failure of the motors driving the left and right wheels. Specially, the effect of the LOE actuator fault is described by:

$$V_r{}^a = V_r{}^c + \beta\,(t - T_r{}^\alpha)\,\alpha_r\;\; V_r{}^c \quad (2)$$

$$V_l{}^a = V_l{}^c + \beta\,(t - T_l{}^\alpha)\,\alpha_l\, V_l{}^c \quad (3)$$

where $V_r{}^a$ and $V_l{}^a$ are the actual velocities of the right and left wheels respectively as defined in (1), $V_r{}^c$ and $V_l{}^c$ are the commanded velocities for the right and left wheels respectively, α_r and α_l represent the unknown parameters of actuator faults due to loss of effectiveness. The functions $\beta\,(t - T_r{}^\alpha)$ and $\beta\,(t - T_l{}^\alpha)$ describe the fault time profiles of

the actuator faults with T_r^{α} and T_l^{α} representing the unknown time instants when the faults occur. For simplicity, in this research we only consider abrupt faults. In other words, we assume the fault time profiles can be modelled as:

$$\beta\,(t - T_r^{\alpha}) = \begin{cases} 0, & \text{if } t < T_r^{\alpha} \\ 1, & \text{if } t \geq T_r^{\alpha} \end{cases} \tag{4}$$

$$\beta\,(t - T_l^{\alpha}) = \begin{cases} 0, & \text{if } t < T_l^{\alpha} \\ 1, & \text{if } t \geq T_l^{\alpha} \end{cases} \tag{5}$$

Therefore, In the absence of actuator faults in the system (nominal case), the actual velocities of the right and left wheels are represented as:

$$V_r^{a} = V_r^{c} \tag{6}$$

$$V_l^{a} = V_l^{c} \tag{7}$$

In other words, the actual wheel velocities and the commanded velocities are equal to each other, representing a healthy robot.

In the presence of actuator faults (i.e., for $t \geq T_r^{\alpha}$ or $t \geq T_l^{\alpha}$), we have

$$V_r^{a} = \alpha_r V_r^{c} + V_r^{c} \tag{8}$$

$$V_l^{a} = \alpha_l V_l^{c} + V_l^{c} \tag{9}$$

The special case of $\alpha_r \; or \; \alpha_l = 0$ represents a healthy motor for the right or left wheel of the robot. This research focuses on partial loss of effectiveness of the actuators by specifying $-1 < \alpha_r \leq 0$ and $-1 < \alpha_l \leq 0$.

Additionally, we consider constant bias sensor faults in the robotic system described by

$$V_r^{s} = V_r^{a} + \beta\,(t - T_r^{b})\, b_r \tag{10}$$

$$V_l^s = V_l^a + \beta\,(t - T_l^b)\,b_l \tag{11}$$

where V_r^a and V_l^a are the actual velocities for right and left wheels respectively as defined in (1), V_r^s and V_l^s are the measured velocities for the wheels, $\beta\,(t - T_r^b)$ and $\beta\,(t - T_l^b)$ are fault time profile functions of the sensor fault which occur at unknown time instants T_r^b and T_l^b respectively. Again, each fault time profile function is assumed to be a step function representing an abrupt fault. Also, the unknown constants b_r and b_l represent the unknown sensor biases in right and left wheels, respectively.

The objective of this research is to develop adaptive fault parameter estimation and fault-tolerant control algorithms to accommodate the effect of actuator and sensor faults in the robotic system and maintain acceptable control performance even in the presence of the faults under consideration.

4 ADAPTIVE FAULT PARAMETER ESTIMATION ALGORITHM

The unknown fault parameters can be estimated to provide the fault information, which can be used to compensate for the fault effects. The estimation methods can possibly be online or offline. Online parameter estimation methods operate in a real time manner to immediately process new sensor data samples when they become available, whereas offline estimation methods involve two different steps including collecting all the data and processing the data batch together for parameter estimation. Thus, online parameter estimation algorithms are in general more efficient and less computationally expensive when compared to the offline estimation methods.

The design of online adaptive estimation methods involves three consecutive steps:

- Selection of an appropriate parameterization of the plant model, so that the model is linear in the unknown parameters.
- Develop an online estimation model.
- Design and analysis of adaptive laws for estimating the unknown plant parameter vector.

In this research, the fault parameter estimation algorithm is designed by comparing the positions and the heading angle of the robot measured by the Vicon system with the corresponding estimates generated by an adaptive estimation scheme. The estimated fault magnitude is utilized by the controller to adjust the control signal to compensate for the fault effects.

4.1 Sensor bias Adaptive Estimation Algorithm

Sensor faults represent faulty measurement values provided by the sensors. There could be various reasons for the sensor faults including broken contact with the robot or adverse environmental factors, leading to sensor related faults in the robot functioning. Sensor faults in the robot system can possibly degrade the control performance or even lead to stability and safety issues.

Let us consider the robotic system dynamics described in (1) and the sensor fault model given by (10) and (11). In the presence of a sensor bias fault in the encoder measuring the right wheel velocity, the robotic system model can be rewritten as:

$$\dot{x} = \frac{({V_r}^s + {V_l}^s)\, cos\varphi}{2} - b_r \left(\frac{cos\varphi}{2}\right)$$

$$\dot{y} = \frac{({V_r}^s + {V_l}^s)\, sin\varphi}{2} - b_r \left(\frac{sin\varphi}{2}\right) \tag{12}$$

$$\dot{\varphi} = \frac{({V_r}^s - {V_l}^s)}{d} - b_r \left(\frac{1}{d}\right) \tag{13}$$

By using the series-parallel estimation model [2], the adaptive fault parameter estimator is designed as follows:

$$\begin{bmatrix}\dot{\hat{x}}\\ \dot{\hat{y}}\end{bmatrix} = \begin{bmatrix}-a_m(\hat{x}-x)\\ -a_m(\hat{y}-y)\end{bmatrix} + \begin{bmatrix}\frac{({V_r}^s + {V_l}^s)\, cos\varphi}{2}\\ \frac{({V_r}^s + {V_l}^s)\, sin\varphi}{2}\end{bmatrix} - \hat{b}_r^1 \begin{bmatrix}\left(\frac{cos\varphi}{2}\right)\\ \left(\frac{sin\varphi}{2}\right)\end{bmatrix} \tag{14}$$

$$\dot{\hat{\varphi}} = -a_m(\hat{\varphi}-\varphi) + \frac{({V_r}^s - {V_l}^s)}{d} - \hat{b}_r^2 \left(\frac{1}{d}\right) \tag{15}$$

where $a_m > 0$ is a design constant, $\hat{x}$, $\hat{y}$ and $\hat{\varphi}$ are the estimated robot position and heading angle respectively, $\hat{b}_r^1$ and $\hat{b}_r^2$ are the estimated sensor bias generated based on the robotic position and heading angle, respectively.

Based on the Lyapunov stability theory [2], the adaptive algorithms for the two adaptive fault estimators given above are designed as follows:

$$\dot{\hat{b}}_r^1 = \gamma_1 \begin{bmatrix} \left(\frac{-cos\varphi}{2}\right) \\ \left(\frac{-sin\varphi}{2}\right) \end{bmatrix}' \begin{bmatrix} (x - \hat{x}) \\ (y - \hat{y}) \end{bmatrix} \tag{16}$$

$$\dot{\hat{b}}_r^2 = \gamma_2 \left(-\frac{1}{d}\right)(\varphi - \hat{\varphi}) \tag{17}$$

where $\gamma_1 > 0$ and $\gamma_2 > 0$ are the learning rates. Additionally, $(x - \hat{x})$ and $(y - \hat{y})$ are position estimation errors and $(\varphi - \hat{\varphi})$ is the heading angle estimation error, respectively.

Analogously, with regard to the sensor bias fault in the left wheel encoder, the adaptive fault estimators and the adaptive algorithms are designed as:

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{y}} \end{bmatrix} = \begin{bmatrix} -a_m(\hat{x} - x) \\ -a_m(\hat{y} - y) \end{bmatrix} + \begin{bmatrix} \frac{(V_r{}^s + V_l{}^s)\, cos\varphi}{2} \\ \frac{(V_r{}^s + V_l{}^s)\, sin\varphi}{2} \end{bmatrix} - \hat{b}_l^1 \begin{bmatrix} \left(\frac{cos\varphi}{2}\right) \\ \left(\frac{sin\varphi}{2}\right) \end{bmatrix} \tag{18}$$

$$\dot{\hat{\varphi}} = -a_m(\hat{\varphi} - \varphi) + \frac{(V_r{}^s - V_l{}^s)}{d} + \hat{b}_l^2 \left(\frac{1}{d}\right) \tag{19}$$

$$\dot{\hat{b}}_l^1 = \gamma_1 \begin{bmatrix} \left(\frac{-cos\varphi}{2}\right) \\ \left(\frac{-sin\varphi}{2}\right) \end{bmatrix}' \begin{bmatrix} (x - \hat{x}) \\ (y - \hat{y}) \end{bmatrix} \tag{20}$$

$$\dot{\hat{b}}_l^2 = \gamma_2 \left(\frac{1}{d}\right)(\varphi - \hat{\varphi}) \tag{21}$$

where $a_m > 0$ is a design constant, $\hat{x}$, $\hat{y}$ and $\hat{\varphi}$ are the estimated robot position and heading angle, respectively, ($\hat{b}_l^1$ and $\hat{b}_l^2$) are the estimated sensor bias generated based on the robotic position and heading angle, respectively. Additionally, γ_1 and γ_2 are the learning rates, $(x - \hat{x})$ and $(y - \hat{y})$ are position estimation errors and $(\varphi - \hat{\varphi})$ is the heading angle estimation error, respectively.

Note that the regressor term $\frac{1}{d}$ in the adaptive algorithms (17) and (21) has opposite signs. As a result, in the presence of a sensor fault in the right wheel encoder, the matched estimator described in (14) and (15) will provide the correct estimation, while the unmatched estimators (18) and (19) designed for the sensor bias in the left wheel encoder will generate estimators with different signs.

Based on this observation, the final estimated values for sensor bias b_r and b_l are designed as:

$$\hat{b}_r = \frac{\hat{b}_r^1 + \hat{b}_r^2}{2} \tag{22}$$

$$\hat{b}_l = \frac{\hat{b}_l^1 + \hat{b}_l^2}{2} \tag{23}$$

where, $\hat{b}_r$ and $\hat{b}_l$ are the final sensor bias estimates that will be used for fault accommodation, $\hat{b}_r^1$ and $\hat{b}_r^2$ are the estimates provided by estimators (14) and (15), respectively. Similarly, $\hat{b}_l^1$ and $\hat{b}_l^2$ are generated by estimators (18) and (19).

Remarks: In the presence of a bias b_r is in the right wheel encoder, both $\hat{b}_r^1$ and $\hat{b}_r^2$ will converge to values around the actual value of b_r. Therefore, the average $\hat{b}_r$ would provide a robust estimation around its true value. However, the estimates $\hat{b}_l^1$ and $\hat{b}_l^2$, generated by the unmatched estimators have opposite signs. As a result, the average $\hat{b}_l$ given by (23) would converge to a value around zero. Hence, the sensor fault can be successfully isolated.

4.2 Loss of Effectiveness Adaptive Estimation Algorithm

Let us consider the robotic system dynamics described in (1) and the LOE actuator fault model given by (2) and (3). In the presence of a loss of effective fault in the motor driving the right wheel, the robotic system model can be rewritten as:

$$\dot{x} = \frac{(V_r^c + V_l^c)\, cos\varphi}{2} + \alpha_r \left(\frac{V_r^c cos\varphi}{2}\right)$$

$$\dot{y} = \frac{(V_r^c + V_l^c)\, sin\varphi}{2} + \alpha_r \left(\frac{V_r^c sin\varphi}{2}\right) \tag{24}$$

$$\dot{\varphi} = \frac{(V_r^c + V_l^c)}{d} + \alpha_r \left(\frac{V_r^c}{d}\right) \tag{25}$$

By using series-parallel adaptive estimation model [2], the adaptive fault parameter estimator is designed as follows:

$$\begin{bmatrix}\dot{\hat{x}}\\ \dot{\hat{y}}\end{bmatrix} = \begin{bmatrix}-a_m(\hat{x}-x)\\ -a_m(\hat{y}-y)\end{bmatrix} + \begin{bmatrix}\frac{(V_r^c+V_l^c)\,cos\varphi}{2}\\ \frac{(V_r^c+V_l^c)\,sin\varphi}{2}\end{bmatrix} + \hat{\alpha}_r^1 \begin{bmatrix}\left(\frac{V_r^c cos\varphi}{2}\right)\\ \left(\frac{V_r^c sin\varphi}{2}\right)\end{bmatrix} \quad (25)$$

$$\dot{\hat{\varphi}} = -a_m(\hat{\varphi}-\varphi) + \frac{(V_r^a - V_l^a)}{d} + \hat{\alpha}_r^2 \left(\frac{V_r^c}{d}\right) \quad (26)$$

where $a_m > 0$ is a design constant, $\hat{x}$, $\hat{y}$ and $\hat{\varphi}$ are the estimated robot position and heading angle, respectively, $\hat{\alpha}_r^1$ and $\hat{\alpha}_r^2$ are the estimates of the loss of effectiveness fault parameter generated based on the robotic position and heading angle, respectively.

Based on the Lyapunov stability theory [2], the adaptive algorithms for the two adaptive fault estimators given above are designed as:

$$\dot{\hat{\alpha}}_r^1 = \gamma_1 \begin{bmatrix}\left(\frac{V_r^c cos\varphi}{2}\right)\\ \left(\frac{V_r^c sin\varphi}{2}\right)\end{bmatrix}' \begin{bmatrix}(x-\hat{x})\\ (y-\hat{y})\end{bmatrix} \quad (27)$$

$$\dot{\hat{\alpha}}_r^2 = \gamma_2 \left(\frac{V_r^c}{d}\right)(\varphi - \hat{\varphi}) \quad (28)$$

where $\gamma_1 > 0$ and $\gamma_2 > 0$ are the learning rates. Additionally, $(x-\hat{x})$ and $(y-\hat{y})$ are position estimation errors and $(\varphi - \hat{\varphi})$ is the heading angle estimation error, respectively.

Analogously, with regard to the LOE actuator fault in the left wheel motor, the adaptive fault estimators are designed as:

$$\begin{bmatrix}\dot{\hat{x}}\\ \dot{\hat{y}}\end{bmatrix} = \begin{bmatrix}-a_m(\hat{x}-x)\\ -a_m(\hat{y}-y)\end{bmatrix} + \begin{bmatrix}\frac{(V_r^c + V_l^c)\,cos\varphi}{2}\\ \frac{(V_r^c + V_l^c)\,sin\varphi}{2}\end{bmatrix} + \hat{\alpha}_l^1 \begin{bmatrix}\left(\frac{V_l^c cos\varphi}{2}\right)\\ \left(\frac{V_l^c sin\varphi}{2}\right)\end{bmatrix} \quad (29)$$

$$\dot{\hat{\varphi}} = -a_m(\hat{\varphi}-\varphi) + \frac{(V_r^c - V_l^c)}{d} - \hat{\alpha}_l^2 \left(\frac{V_l^c}{d}\right) \quad (30)$$

$$\dot{\hat{\alpha}}_l^1 = \gamma_1 \begin{bmatrix}\left(\frac{V_l^c cos\varphi}{2}\right)\\ \left(\frac{V_l^c sin\varphi}{2}\right)\end{bmatrix}' \begin{bmatrix}(x-\hat{x})\\ (y-\hat{y})\end{bmatrix} \quad (31)$$

$$\dot{\hat{\alpha}}_l^2 = \gamma_2 \left(\frac{-V_l^c}{d}\right)(\varphi - \hat{\varphi}) \quad (32)$$

where $a_m > 0$ is a design constant, $\hat{x}$, $\hat{y}$ and $\hat{\varphi}$ are the estimated robot position and heading angle, respectively, $\hat{\alpha}_l^1$ and $\hat{\alpha}_l^2$ are the estimated loss of effectiveness fault parameter generated based on the robotic position and heading angle, respectively. Additionally, γ_1 and γ_2 are the learning rates, $(x-\hat{x})$ and $(y-\hat{y})$ are position estimation errors, and $(\varphi - \hat{\varphi})$ is the heading angle estimation error.

Note that the regressor term $\frac{1}{d}$ in the adaptive estimators (26) and (30) has opposite signs. As a result, in the presence of a LOE actuator fault in right wheel motor, the matched estimator described in (25) and (26) will provide the correct estimation, while the estimates, $\hat{\alpha}_l^1$ and $\hat{\alpha}_l^2$ generated by the unmatched estimators (29) and (30) designed for the LOE actuator fault in the left wheel motor will have different signs.

Based on this observation, the final estimated values for the LOE fault parameter α_r and α_l are designed as:

$$\hat{\alpha}_r = \frac{\hat{\alpha}_r^1 + \hat{\alpha}_r^2}{2} \tag{31}$$

$$\hat{\alpha}_l = \frac{\hat{\alpha}_l^1 + \hat{\alpha}_l^2}{2} \tag{32}$$

Remarks: In the presence of a LOE fault in the right wheel motor, both $\hat{\alpha}_r^1$, $\hat{\alpha}_r^2$ will converge to the values around α_r . Therefore, the average $\hat{\alpha}_r$ would provide a robust estimation around its true value. However, the estimates $\hat{\alpha}_l^1$ and $\hat{\alpha}_l^2$ generated by the unmatched estimators (29) and (30) have opposite signs. As a result, the average $\hat{\alpha}_l$ given by (32) would converge to a value around zero. Hence, the LOE actuator fault can be successfully isolated.

5 EXPERIMENTAL RESULTS

In this chapter, the implementation and evaluation of adaptive fault-tolerant control algorithms are described. Real-time experimental results are shown to illustrate the effectiveness of the algorithm.

5.1 MATLAB Implementation

The architecture of the fault-tolerant control scheme is shown in Figure 6. It consists of four main components: trajectory generation, PID controller, the ground vehicle including the actuators and sensors, and the adaptive fault parameter estimation mechanism. The fault parameter estimates are provided to the PID controller for accommodating the fault effects.

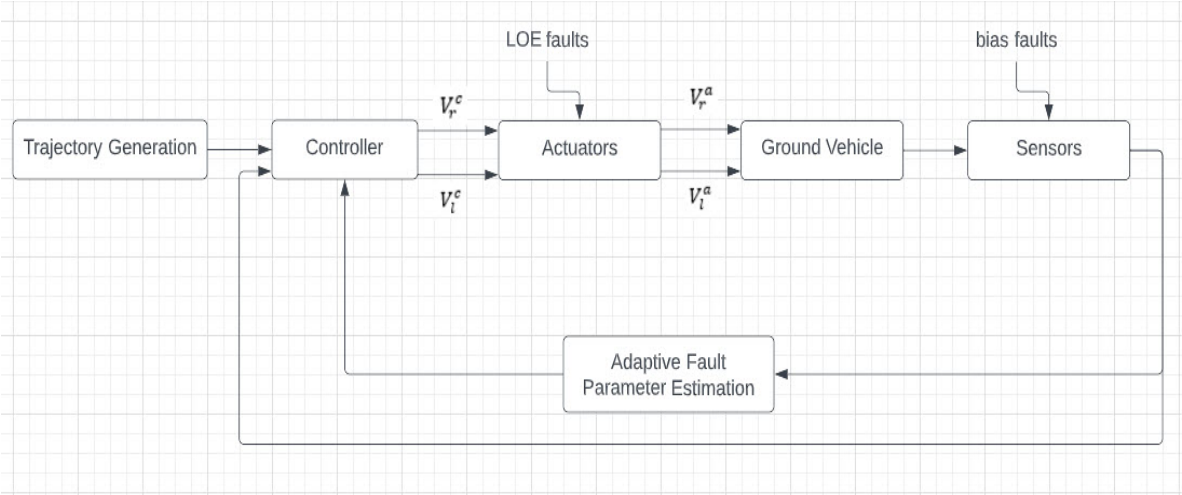


Figure 6: Schematic diagram of Simulink model designed for the thesis

A simple trajectory generation algorithm is implemented, generating way points leading to a circular trajectory. A simple proportional controller is used as the controller component.

The algorithms implemented in the component of adaptive fault parameter estimation are described in Section 4.

The LOE faults and sensor bias faults are simulated based on (2)-(3) and (10)-(11), respectively. All the algorithms were implemented using MATLAB/Simulink enhanced with the Quarc real-time control software from Quanser [5].

5.2 Nominal control performance (Without faults)

This is the condition when the robot operates without any faults occurring to the system. In other words, the robot is in the fault-free condition where all the system components are working properly. The nominal PID controller is responsible for generating desired vehicle velocity and heading angle commands for the robot to move in a circular trajectory provided by the trajectory generation block. The velocity and heading angle commands specifies the desired velocities for the right and left wheels of the robot. The tracking performance of the robot is shown in Figure 7 and Figure 8.

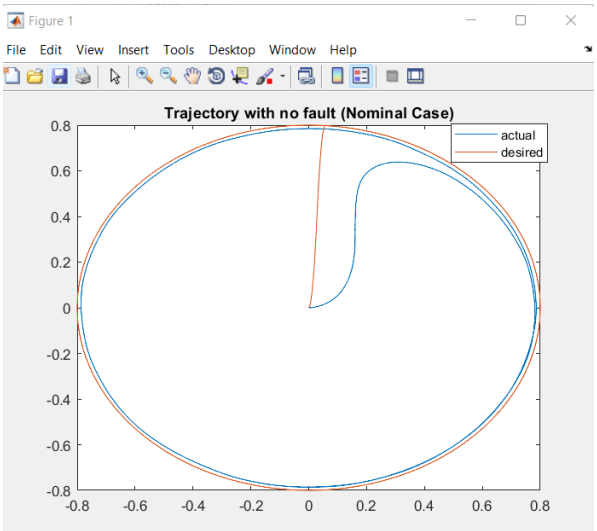


Figure 7: Tracking trajectory with the nominal controller for fault-free case.

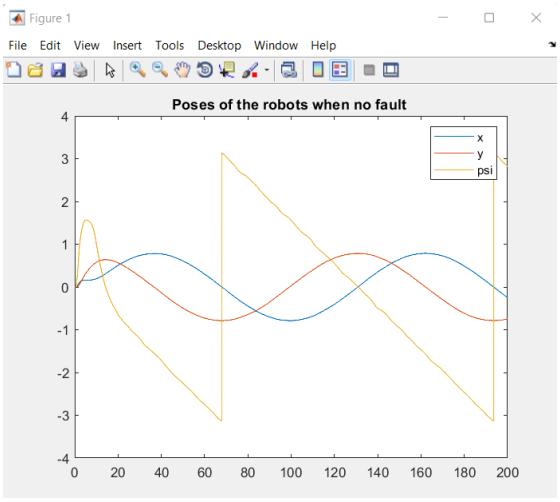


Figure 8: Poses of the robot (x, y and psi angles)

As it can be seen from Figure 7, the robot tracks the desired trajectory very closely in the fault-free condition. The pose of the robot is also shown in Figure 8 for the nominal case where at approximately *t=70 seconds* and *t=190 seconds*, the heading angle switches from $-\pi$ to π to satisfy $-\pi \leqslant \varphi \leq \pi$.

5.3 Adaptive estimation with simulated faults

To observe the effects of different types of faults on the robot, faults are injected artificially through the real-time control software to simulate the fault effects. The following fault scenarios are considered: (1) fault-free; (2) single sensor fault in one of the encoders; (3) sensor faults in both encoders of the robot simultaneously; (4) loss of effectiveness in a single wheel motor; (5) loss of effectiveness in two robot wheels; (6) simultaneous actuator (LOE) and sensor faults in the robotic system, and (7) simultaneous faults in both the two motors and two encoders.

We will be discussing every case in detail, explaining the experimental results of the fault-tolerant control scheme. The trajectories of the ground robot with and without fault accommodation will be compared, demonstrating the effectiveness of algorithms.

5.3.1 Fault-Free Condition

In this case all the control system components are free of faults. The experimental results of the adaptive fault parameter estimation algorithm are shown in Figure 9-10. It can be seen from Figure 9 that the sensor bias estimates both reach values around zero. Additionally, it is observed that LOE estimates have a very small value of approximately -0.05 instead of zero, which represents about 5% LOE. Note that this is caused due to modelling uncertainties in the system. Therefore, all the fault parameter estimates are very small, indicating the estimation results at the fault-free conditions are satisfactory.

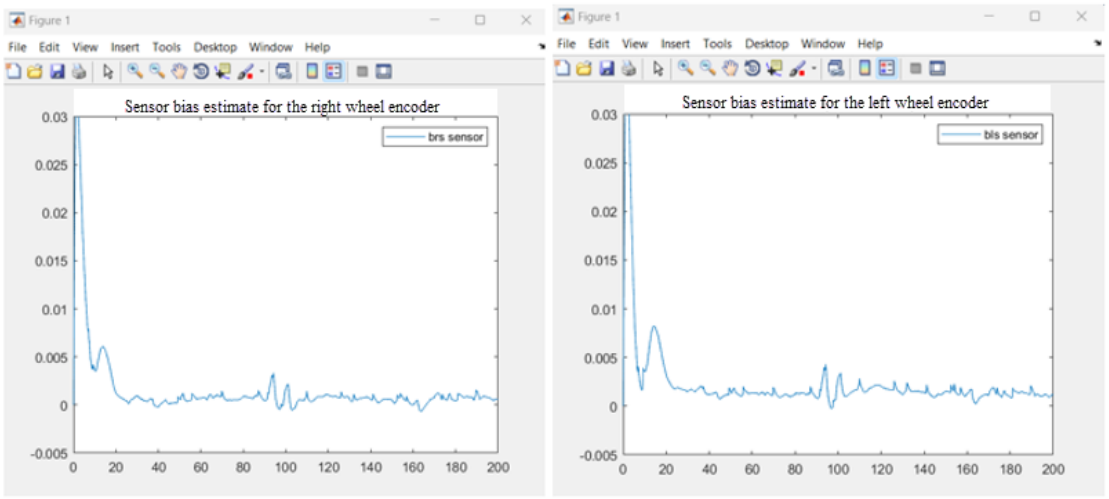


Figure 9 (a) **Figure 9 (b)**

Figure 9a and Figure 9b: Adaptive sensor bias estimates for fault-free condition.

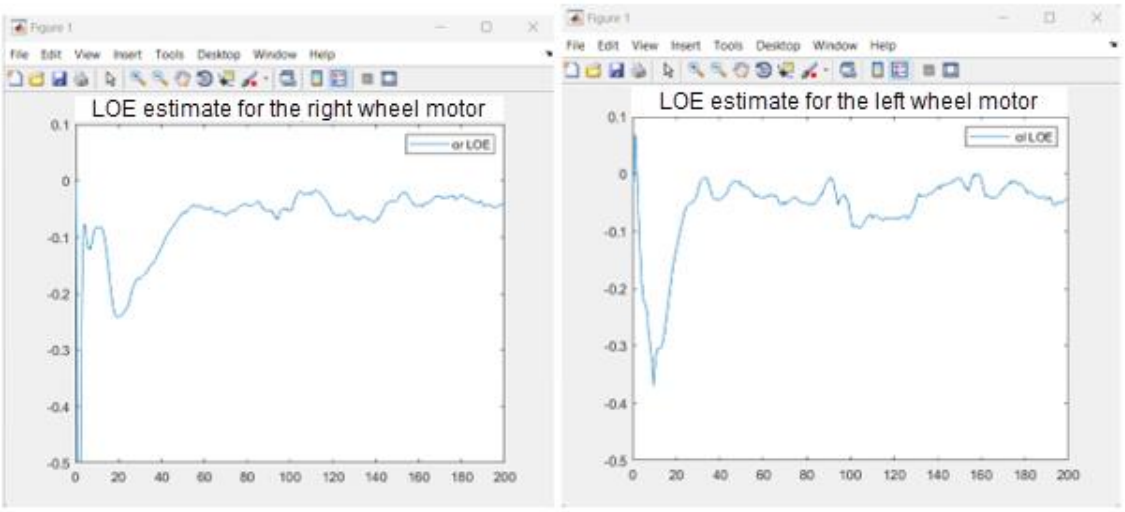


Figure 10 (a) **Figure 10 (b)**

Figure 10a and Figure 10b: Adaptive actuator LOE fault parameter estimates for fault-free condition.

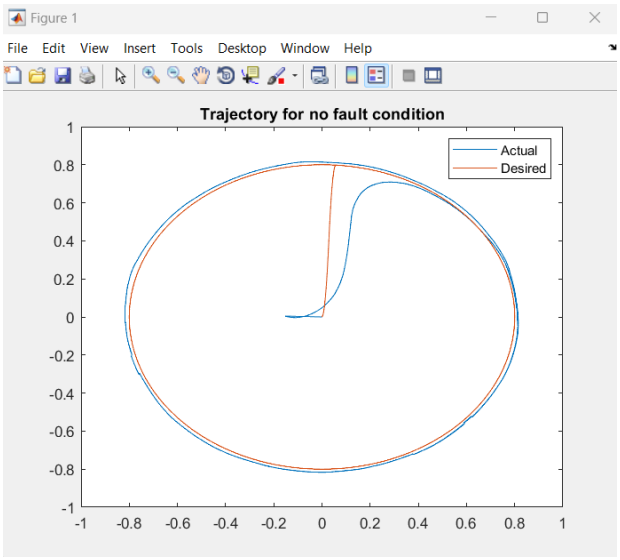


Figure 11: Trajectory performance of the fault-tolerant control scheme for fault-free case

Furthermore, Figure 11 shows the trajectory tracking performance with the fault-tolerant control scheme, where the controller commands to the motor are adjusted using the fault parameter estimates, as shown in Figure 6. As it can be seen from Figure 11 and Figure 7,

the tracking performances are very similar, which illustrates the effectiveness of the fault-tolerant control scheme in fault-free condition.

5.3.2 Sensor faults

1. Single sensor fault- sensor bias b_r= 0.07

This is the case when one of the encoder sensors of the robot is subjected to a bias fault. Specifically, the fault is injected at *t=50 seconds* to the right wheel encoder of the robot with a value of 0.07.

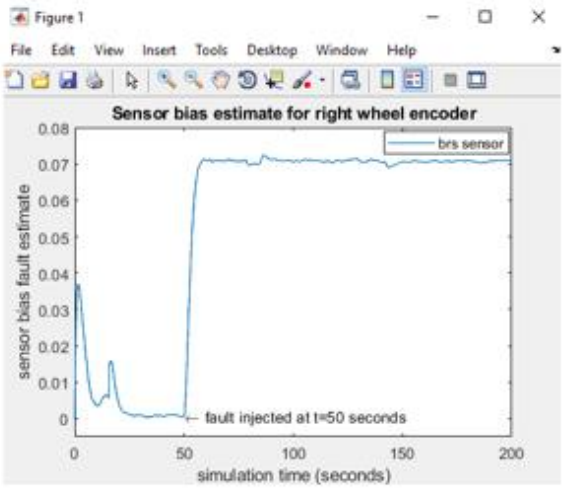


Figure 12 (a)

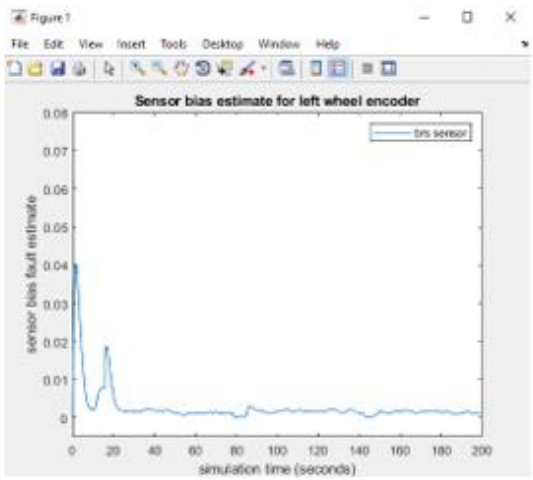


Figure 12 (b)

Figure 12a and Figure 12b: Adaptive sensor bias estimates for single sensor fault in the right wheel encoder.

Figure 12 shows the estimated sensor bias generated by the adaptive algorithm. It can be seen that the estimated sensor bias for the right wheel encoder converges to a value of 0.07 while the bias estimates for the left wheel is approximately around zero.

Additionally, the actuator LOE fault parameter estimates are given in Figure 13. As we can see, the estimates take very small values that are similar to the fault-free case shown in Figure 10. Thus, the sensor bias fault in the right wheel encoder can be successfully identified.

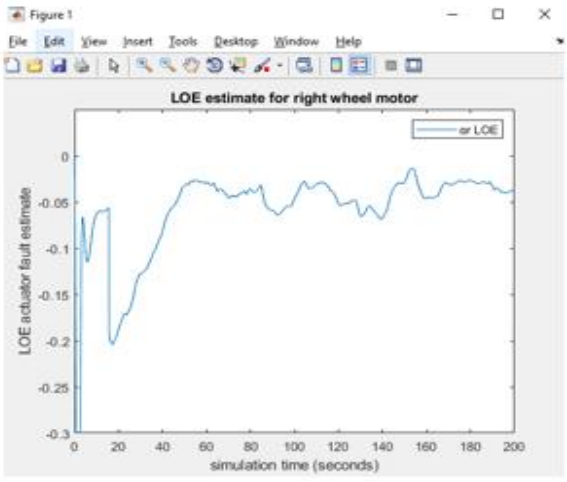


Figure 13 (a)

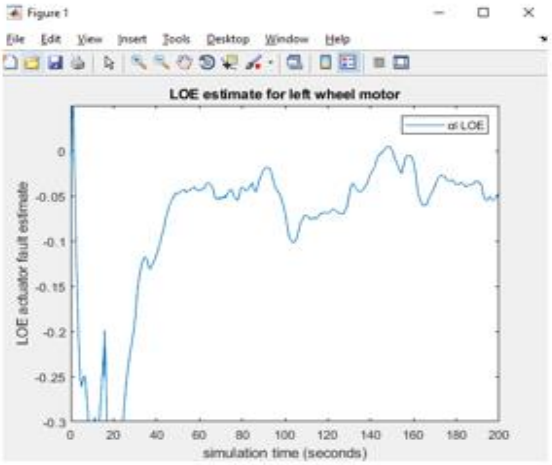
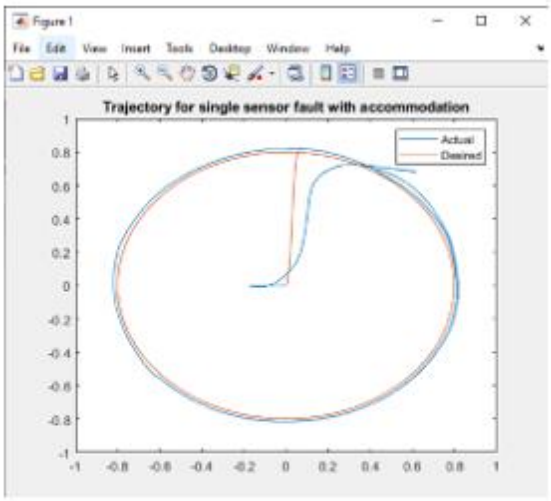


Figure 13 (b)

Figure 13a and Figure 13b: Adaptive actuator LOE fault parameter estimates for single sensor fault in right wheel encoder.



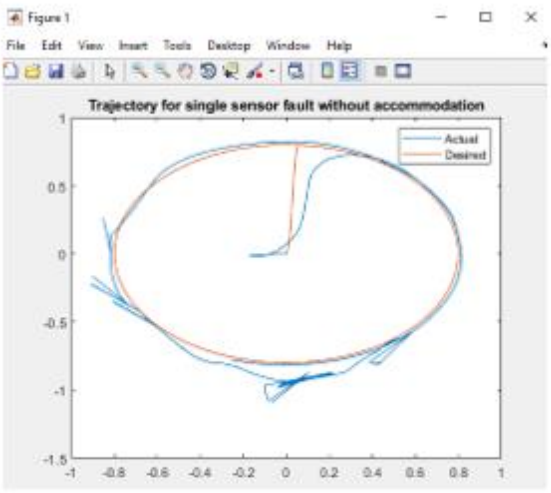
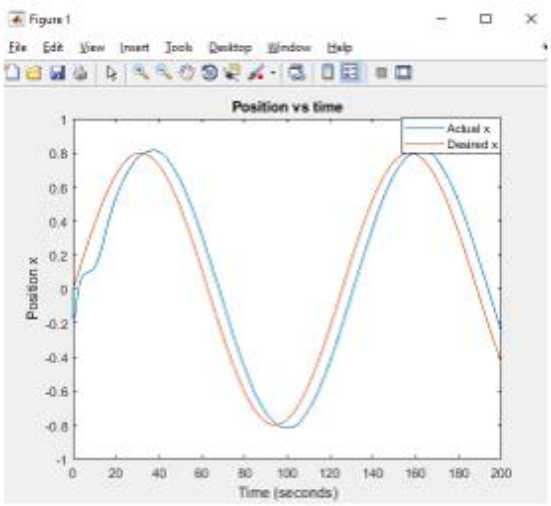


Figure 14 (a) **Figure 14 (b)**

Figure 14a and Figure 14b: Circular trajectory tracking for single sensor fault condition with and without accommodation.



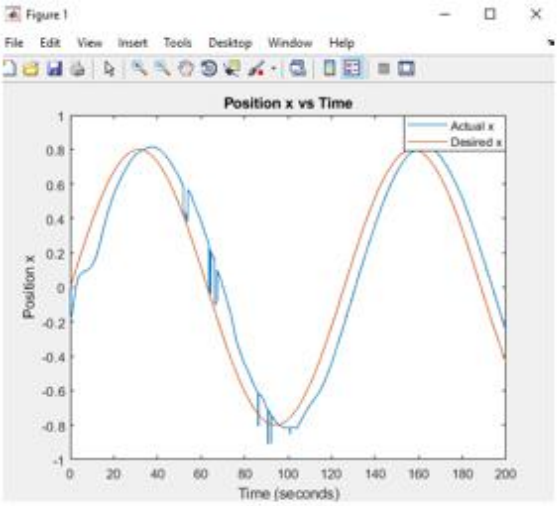


Figure 15 (a) **Figure 15 (b)**

Figure 15a and Figure 15b: Position x with respect to time with and without accommodation for single sensor fault in right wheel encoder.

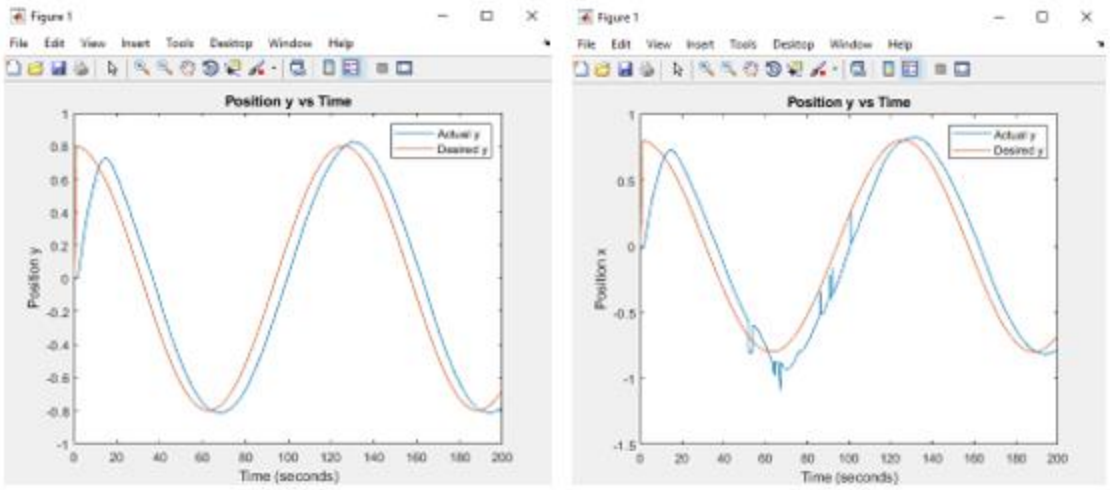


Figure 16 (a) **Figure 16 (b)**

Figure 16a and Figure 16b: Position y with respect to time with and without accommodation for single sensor fault in right wheel encoder.

Figure 14 compares the fault-tolerant control tracking performance with the case without fault accommodation. The red line in the plot shows the desired trajectory, and blue line shows the actual trajectory. These signals are plotted for a time interval of *t=200 seconds.* As can be seen from Figure 14(b), the robot's actual trajectory without accommodation is way off the desired reference trajectory. In contrast, it is observed from Figure 14(a) that the tracking performance is successfully recovered, illustrating the effectiveness of the adaptive fault accommodation method.

Figure 15 shows the comparison between x axis plots with respect to time for cases with and without accommodation. The red line in the plot shows the desired trajectory, and blue line shows the actual trajectory. As it can be seen from Figure 15(b), The trajectory without accommodation shows fluctuations in the actual path due to the presence of fault in the encoder of the robot. In contrast from Figure 15(a) the tracking performance is smooth proving the effectiveness of the adaptive algorithm. Similarly, from Figure 16 shows

comparison of fault-tolerant control performance taking y axis with respect to time to shows the effectiveness of the control algorithm.

2. Simultaneous sensor faults (b_r=0.08 and b_l=-0.02)

This is the case when both of the encoder sensors of the robot are subjected to a bias fault. Specifically, the faults are injected at *t=50 seconds* and *t=100 seconds* to right wheel and left wheel encoders of the robot, respectively, with each sensor bias being 0.08 and -0.02 respectively.

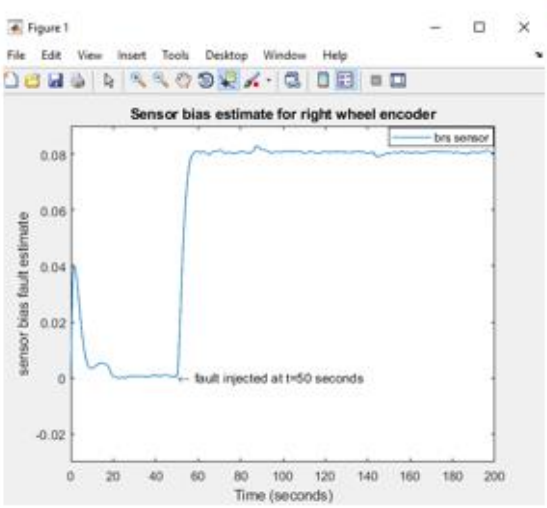


Figure 17 (a)

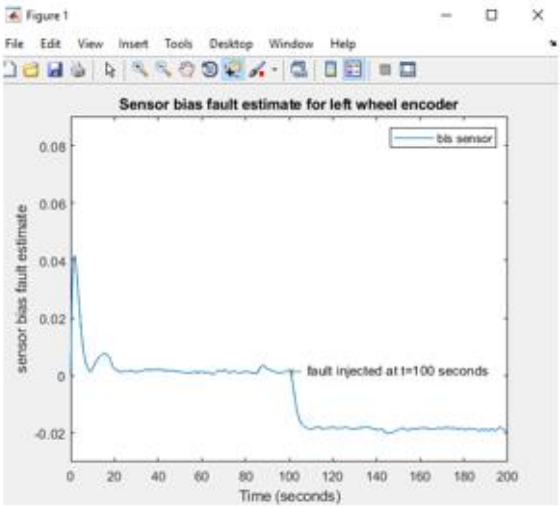
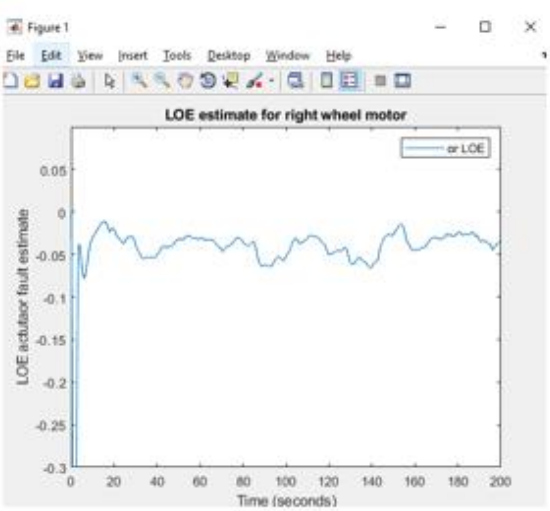


Figure 17 (b)

Figure 17a and Figure 17b: Adaptive sensor bias estimates for sensor fault in both encoders

Figure 17 shows the estimated sensor biases generated by the adaptive algorithm. It can be seen that the estimated sensor biases for the right wheel and left wheel encoders converges to a value of 0.08 and -0.02 respectively. Additionally, the actuator LOE fault parameter estimates are given in Figure 18. As we can see, the estimates take very small values that are similar to the fault-free case shown in Figure 10. Thus, the sensor bias fault in both the right and left wheel encoders can be successfully identified.



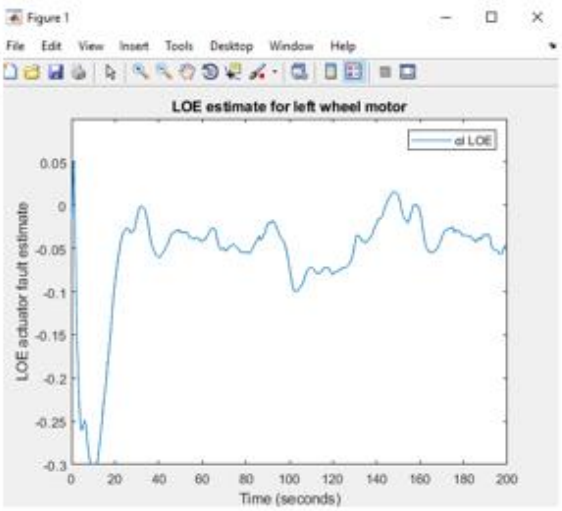
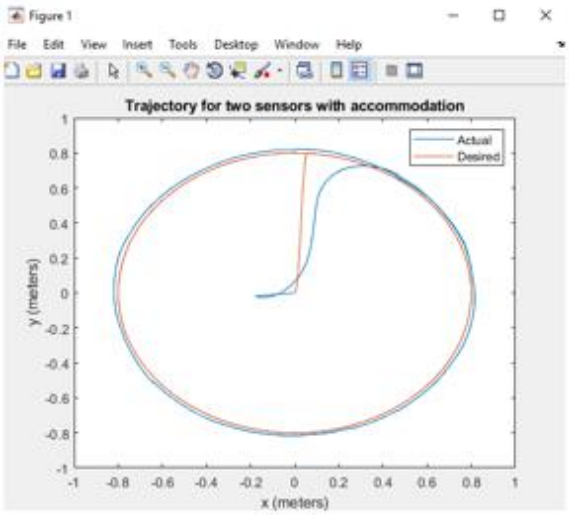


Figure 18 (a) **Figure 18 (b)**

Figure 18a and Figure 16b: Adaptive actuator LOE fault parameter estimates for sensor fault in both encoders.



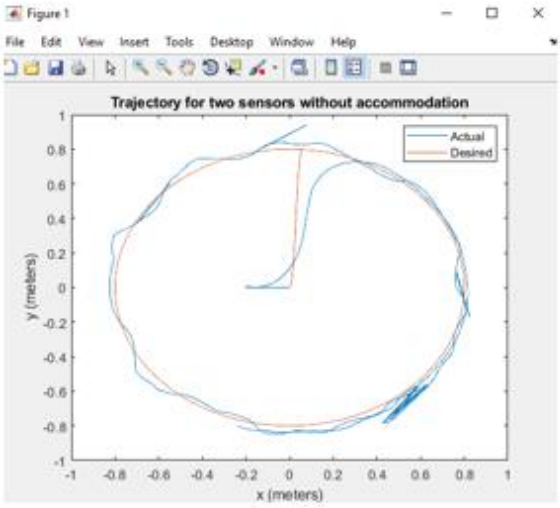


Figure 19 (a) **Figure 19 (b)**

Figure 19a and Figure 19b: Circular trajectory with sensor faults in the encoders of the system.

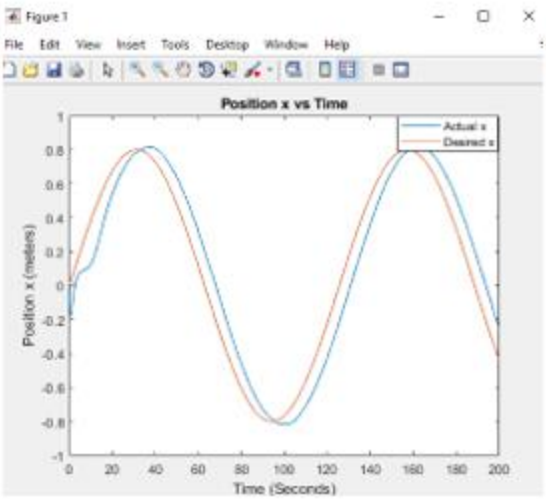


Figure 20 (a)

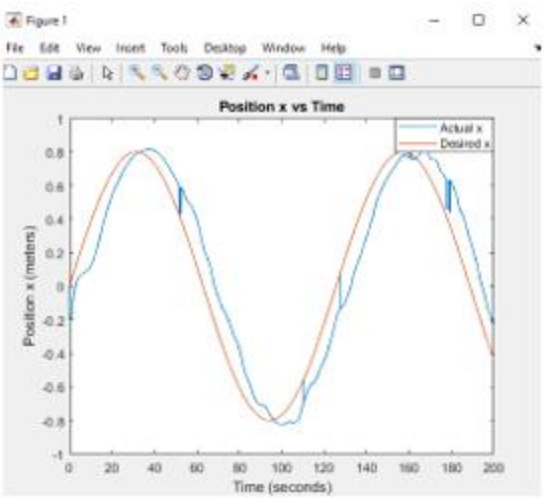


Figure 20 (b)

Figure 20a and Figure 20b: Position x with respect to time with and without accommodation for single sensor fault in right wheel encoder.

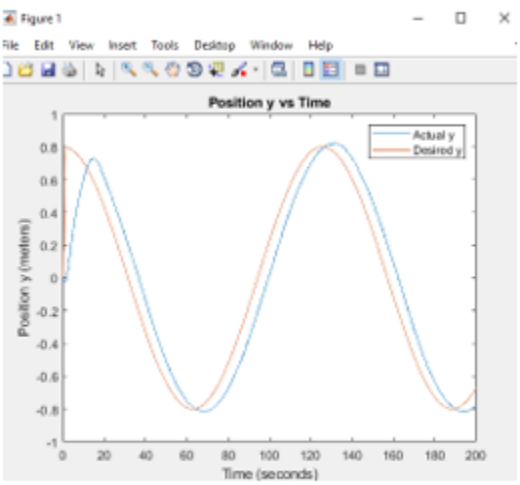


Figure 21 (a)

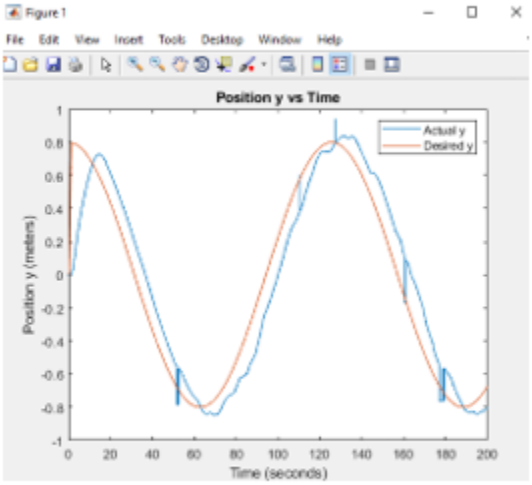


Figure 21 (b)

Figure 21a and Figure 21b: Position y with respect to time with and without accommodation for single sensor fault in right wheel encoder.

Figure 19 compares the fault-tolerant control tracking performance with the case without fault accommodation. The red line in the plot shows the desired trajectory, and blue line shows the actual trajectory. These signals are plotted for a time interval of *t=200 seconds.* As can be seen from Figure 17(b), the robot's actual trajectory without accommodation is way off the desired reference trajectory. In contrast, it is observed from Figure 17(a) that the tracking performance is successfully recovered, by the adaptive fault accommodation method.

Figure 20 shows the comparison between x axis plots with respect to time for cases with and without accommodation. The red line in the plot shows the desired trajectory, and blue line shows the actual trajectory. As it can be seen from Figure 20(b), The trajectory without accommodation shows fluctuations in the actual path due to the presence of fault in the encoder of the robot. In contrast from Figure 20(a) the tracking performance is smooth proving the effectiveness of the adaptive algorithm. Similarly, from Figure 21 shows comparison of fault-tolerant control performance taking y axis with respect to time to shows the effectiveness of the control algorithm.

5.3.3 Actuator faults due to loss of effectiveness

1. Loss of effectiveness- single fault (α_r=-0.6)

This is the case when one of the motors of the robot is subjected to an actuator LOE fault. Specifically, the fault is injected at *t=50 seconds* to the right wheel motor of the robot with a value of -0.6.

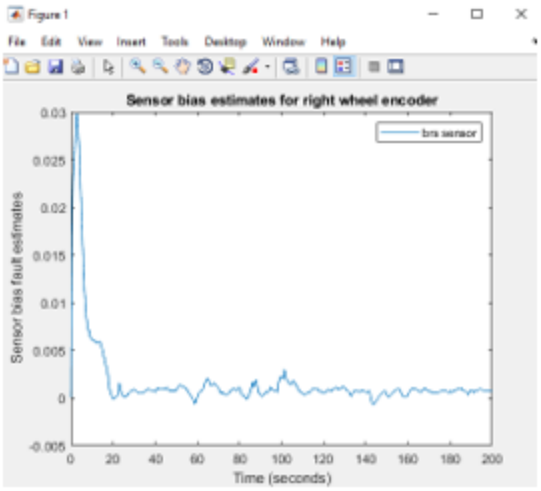


Figure 22 (a)

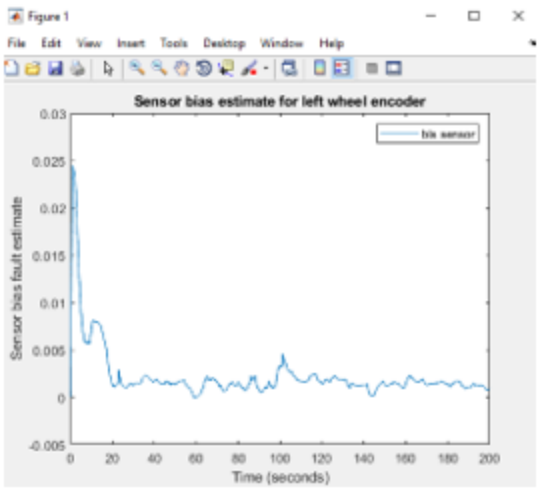
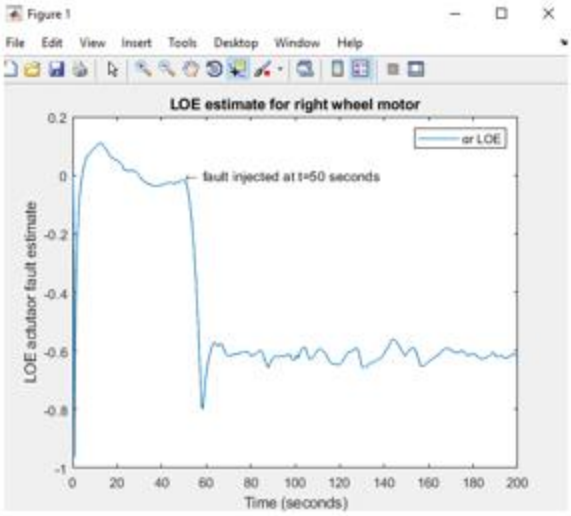


Figure 22 (b)

Figure 22a and Figure 22b: Adaptive sensor bias estimates for loss of effectiveness in single wheel motor.

Figure 22 shows the estimated sensor bias generated by the adaptive algorithm. It can be seen that the estimated sensor bias for the right and left wheels of the robot is approximately zero. Additionally, the actuator LOE fault parameter estimates are given in Figure 23. As we can see, the estimated LOE fault for the right wheel motor converges to a value of approximately -0.6, while the LOE estimates for the left wheel takes very small values that are similar to the fault-free case shown in Figure 10. Thus, the LOE actuator fault in the right wheel motor can be successfully identified.



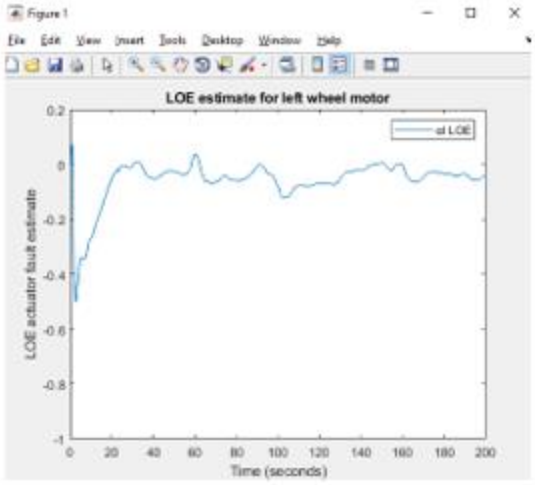
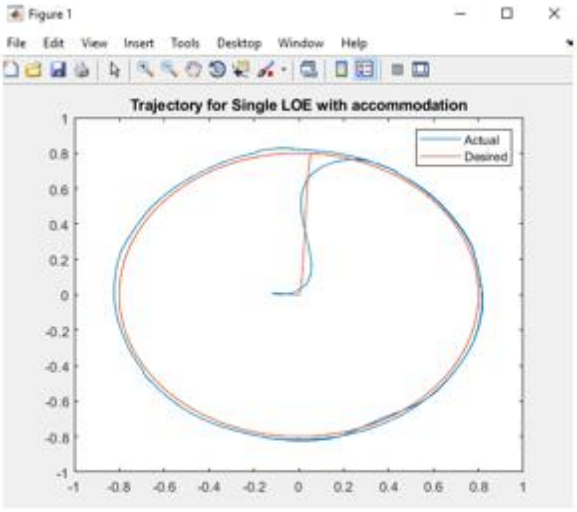


Figure 23(a) **Figure 23(b)**

Figure 23a and Figure 23b: Adaptive actuator LOE fault parameter estimates loss of effectiveness in single wheel motor.



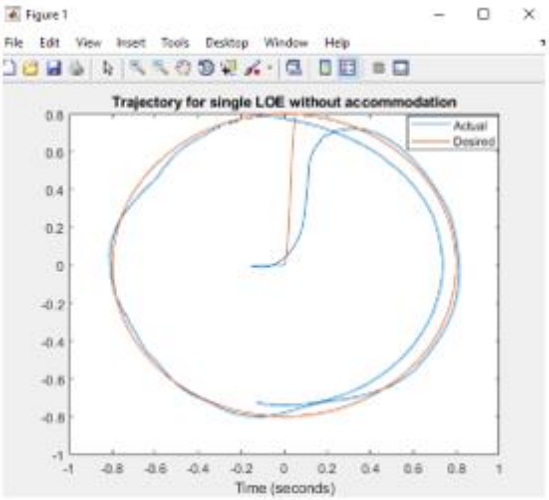


Figure 24 (a) **Figure 24 (b)**

Figure 24a and Figure 24b: Circular trajectory for a single LOE faults in system.

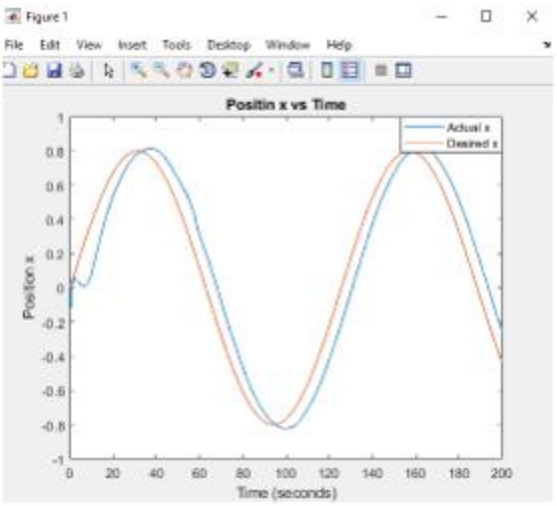


Figure 25 (a)

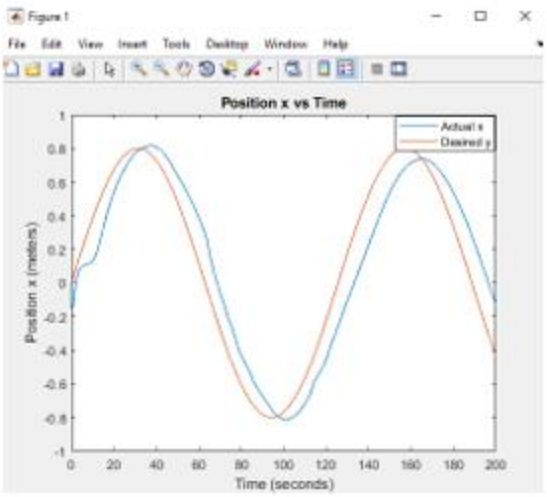


Figure 25 (b)

Figure 25a and Figure 25b: Position x with respect to time with and without accommodation for single LOE fault in right wheel encoder.

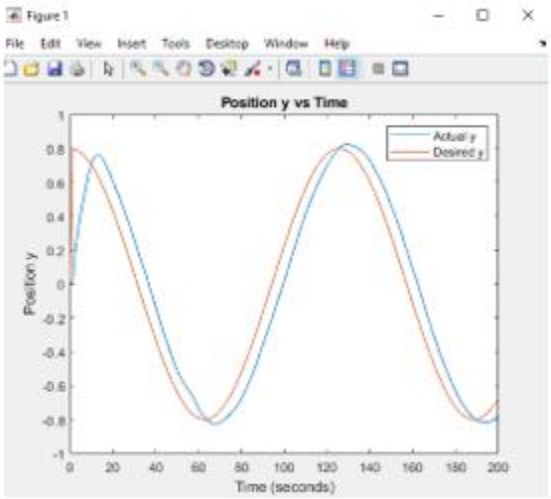


Figure 26 (a)

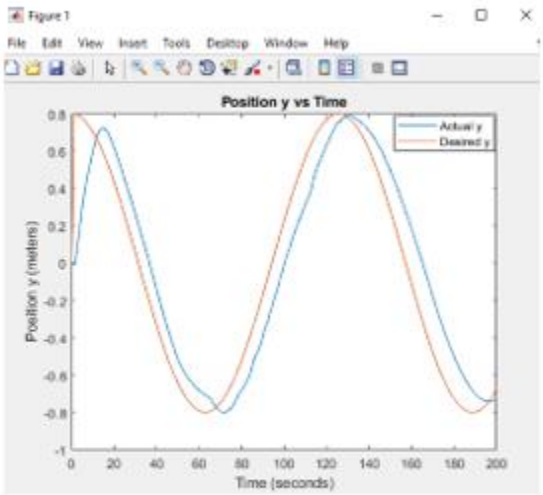


Figure 26 (b)

Figure 26a and Figure 26b: Position y with respect to time with and without accommodation for single LOE fault in right wheel encoder.

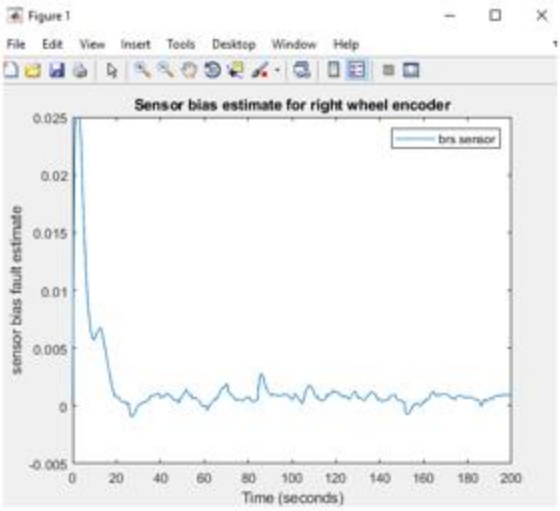
Figure 24 compares the fault-tolerant control tracking performance with the case without fault accommodation. The red line in the plot shows the desired trajectory, and blue line shows the actual trajectory. These signals are plotted for a time interval of *t=200 seconds.*

As can be seen from Figure 24(b), the robot's actual trajectory without accommodation clearly deviates from the desired reference trajectory as a result of fault. In contrast, it is observed from Figure 24(a) that the tracking performance is successfully recovered, illustrating the effectiveness of the adaptive fault accommodation method.

Figure 25 shows the comparison between x axis plots with respect to time for cases with and without accommodation. The red line in the plot shows the desired trajectory, and blue line shows the actual trajectory. As it can be seen from Figure 25(b), The trajectory without accommodation shows fluctuations in the actual path due to the presence of fault in the encoder of the robot. In contrast from Figure 25(a) the tracking performance is smooth proving the effectiveness of the adaptive algorithm. Similarly, from Figure 26 shows comparison of fault-tolerant control performance taking y axis with respect to time to shows the effectiveness of the control algorithm.

2. Loss of effectiveness-simultaneous faults (α_r=-0.4 and $\alpha_l = -0.6$)

This is the case when both the motors of the robot are subjected to an actuator LOE fault. Specifically, the faults are injected at *t=50 seconds* and *t=100 seconds* to the right and left wheels motor of the robot with a value of -0.4 and -0.6, respectively.



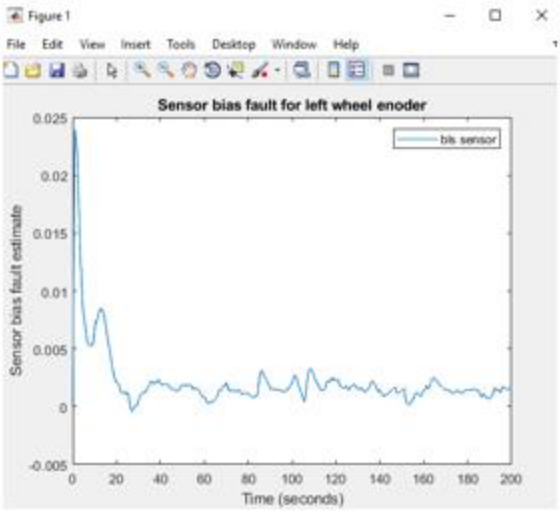
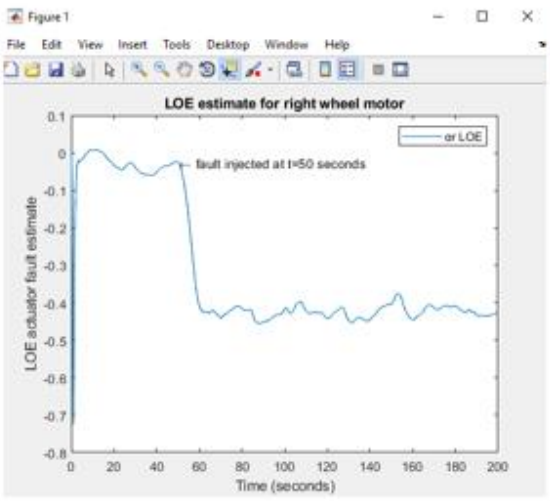


Figure 27 (a) **Figure 27 (b)**

Figure 27a and Figure 27b: Adaptive sensor bias estimates for loss of effectiveness in two robot wheels.

Figure 27 shows the estimated sensor bias generated by the adaptive algorithm. It can be seen that the estimated sensor bias for the right and left wheels of the robot is approximately around zero. Additionally, the actuator LOE fault parameter estimates are given in Figure 28. As we can see, the estimated LOE fault parameters for both the right and the left wheel motors converge to values around -0.4 and -0.6, respectively. Thus, the LOE actuator fault in both the right and left wheel motors can be successfully identified.



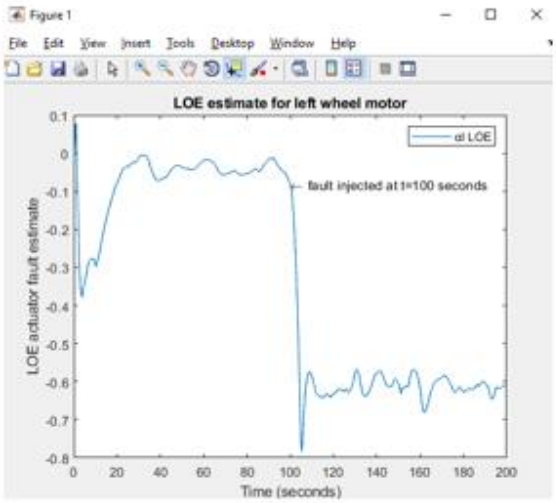
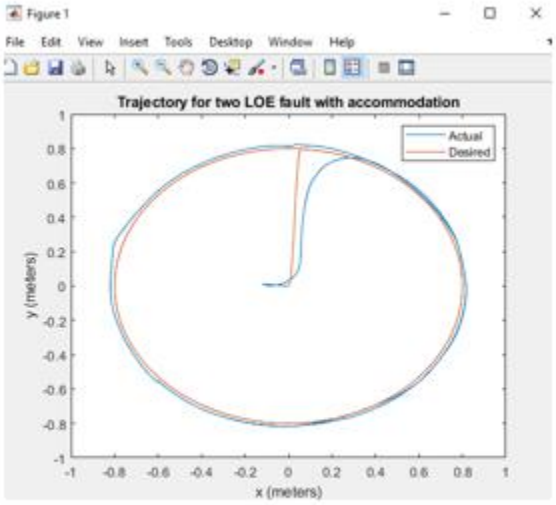


Figure 28 (a) **Figure 28 (b)**

Figure 28a and Figure 28b: Adaptive actuator LOE fault parameter estimates for loss of effectiveness in two robot wheels.



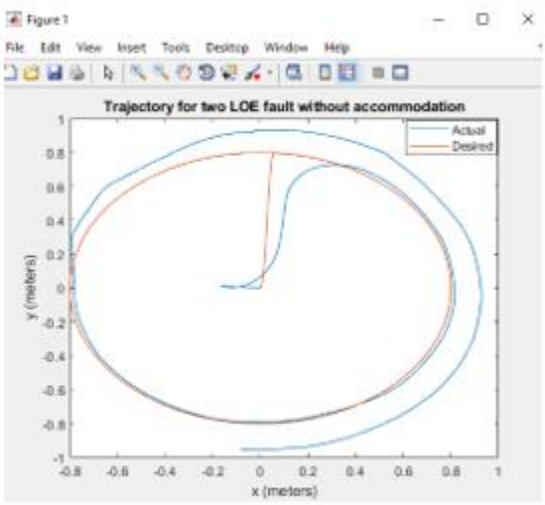


Figure 29 (a) **Figure 29 (b)**

Figure 29a and Figure 29b: Tracking performance of the robot in the presence of LOE of -0.4 in the right wheel and -0.6 in the left wheel of the robot with and without fault accommodation.

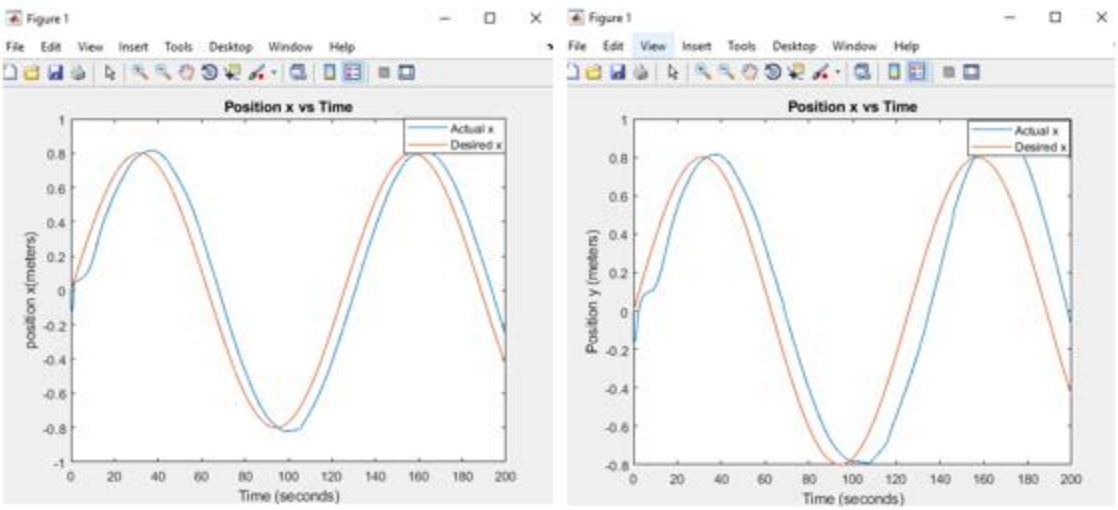


Figure 30 (a) **Figure 30 (b)**

Figure 30a and Figure 30b: Position x with respect to time with and without accommodation for two LOE faults in both wheels of the encoder.

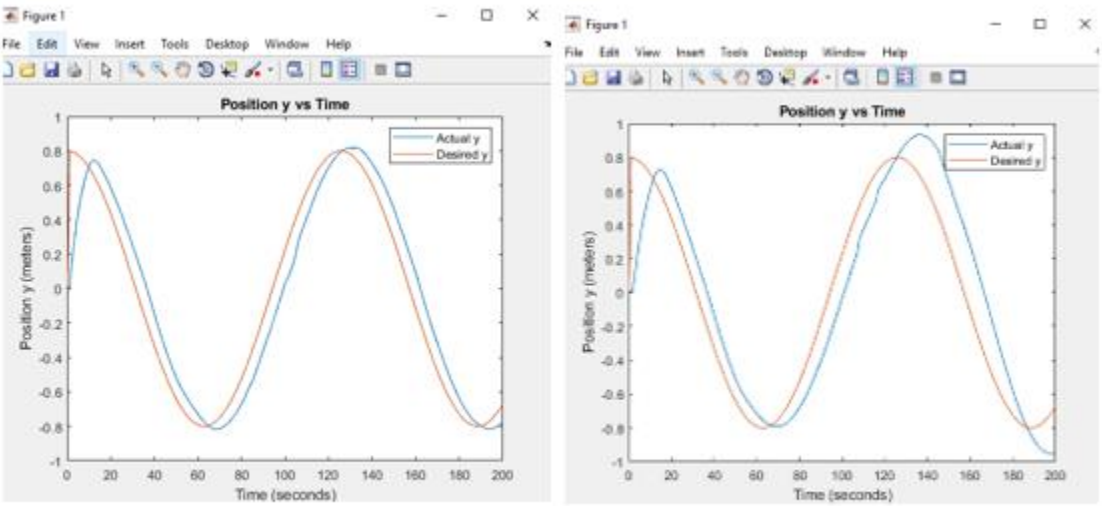


Figure 31 (a) **Figure 31 (b)**

Figure 31a and Figure 31b: Position y with respect to time with and without accommodation for two LOE faults in both wheels of the encoder.

Figure 29 compares the fault-tolerant control tracking performance with the case without fault accommodation. The red line in the plot shows the desired trajectory, and blue line

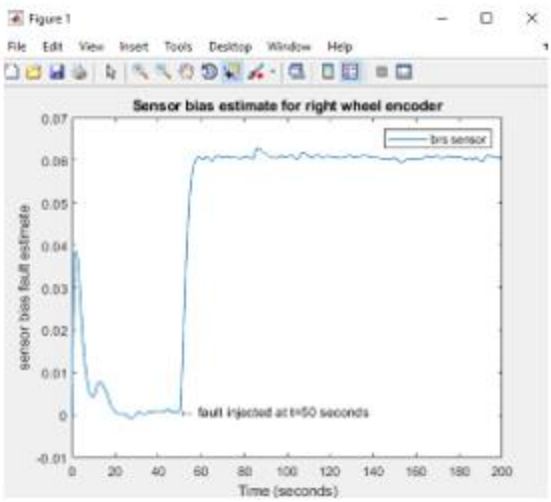
shows the actual trajectory. These signals are plotted for a time interval of *t=200 seconds.* As can be seen from Figure 29(b), the robot's actual trajectory without accommodation significantly degrades as a result of fault occurrence. In contrast, it is observed from Figure 29(a) that the tracking performance is successfully recovered, by the adaptive fault accommodation method.

Figure 30 shows the comparison between x axis plots with respect to time for cases with and without accommodation. The red line in the plot shows the desired trajectory, and blue line shows the actual trajectory. As it can be seen from Figure 30(b), The trajectory without accommodation shows fluctuations in the actual path due to the presence of fault in the encoder of the robot. In contrast from Figure 30(a) the tracking performance is smooth proving the effectiveness of the adaptive algorithm. Similarly, from Figure 31 shows comparison of fault-tolerant control performance taking y axis with respect to time to shows the effectiveness of the control algorithm.

5.3.4 Simultaneous actuator (LOE) and sensor faults

1. Combinations of actuator and sensor faults (b_r=0.07 and α_l=-0.6)

This is the case when one of the motors and one of the encoder sensors of the robot are subjected to simultaneous actuator (LOE) and sensor faults. Specifically, a fault is injected at *t=50 seconds* to the right wheel encoder of the robot with a value of 0.07 and a fault is injected at *t=100 seconds* to the left wheel motor with a value of -0.6.



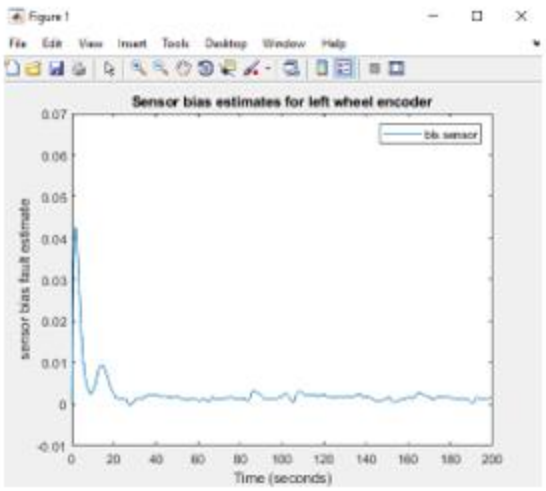


Figure 32(a) **Figure 32 (b)**

Figure 32a and Figure 32b: Adaptive sensor bias estimates for simultaneous actuator (LOE) and sensor faults.

Figure 32 shows the estimated sensor biases generated by the adaptive algorithm. It can be seen that the estimated sensor biases for the right wheel encoder converges to a value of 0.07, while the bias estimate for the left wheel is around zero. Additionally, the actuator LOE fault parameter estimates are given in Figure 33. As we can see, the estimated LOE fault for the right wheel motor converges to a value around -0.6, while the LOE estimates for left wheel takes very small values that are similar to the fault-free case shown in Figure 10. Thus, the LOE actuator fault in the right wheel motor and the sensor bias fault in right wheel encoder can be successfully identified.

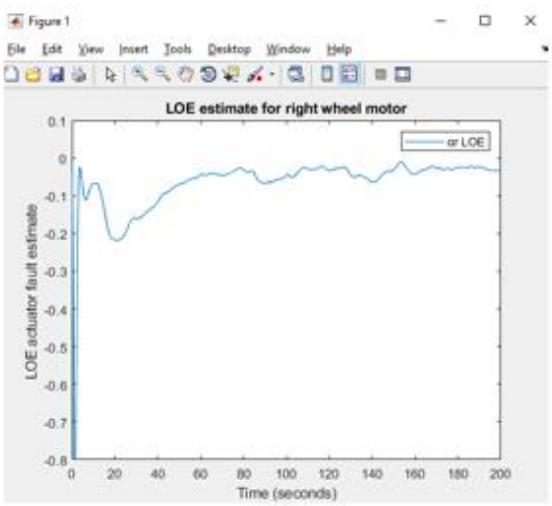


Figure 33 (a)

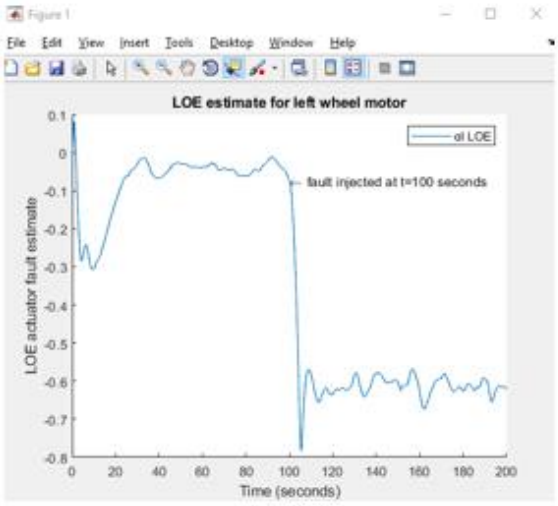


Figure 33 (b)

Figure 33a and Figure 33b: Adaptive actuator LOE fault parameter estimates for simultaneous actuator (LOE) and sensor faults.

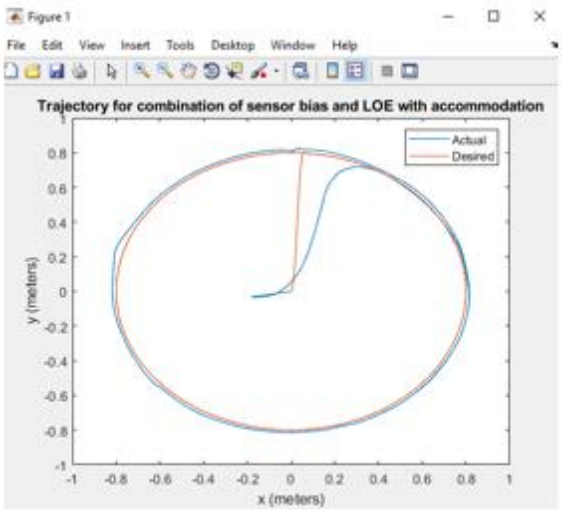


Figure 34 (a)

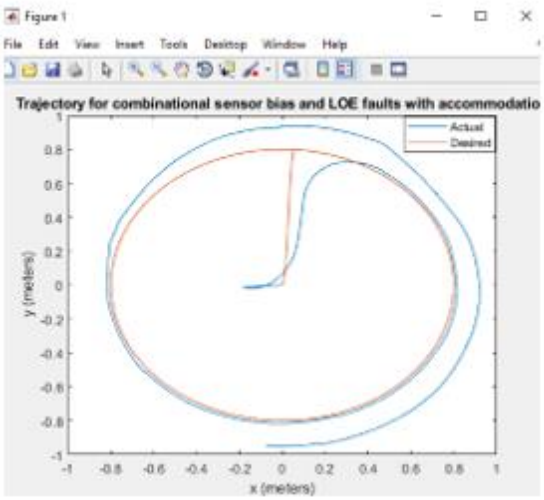


Figure 34 (b)

Figure 34a and Figure 34b: Circular trajectory for combinational sensor and LOE fault with and without accommodation.

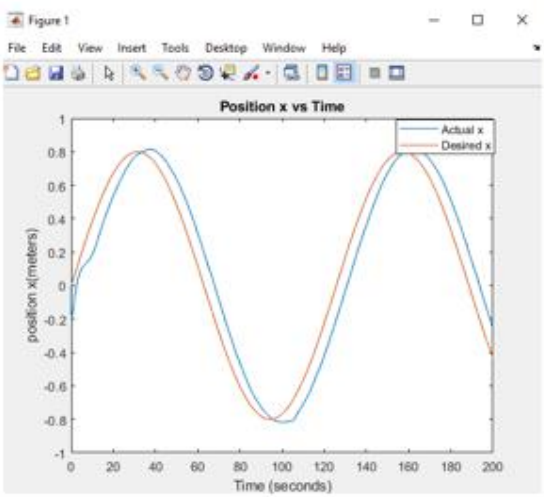


Figure 35 (a)

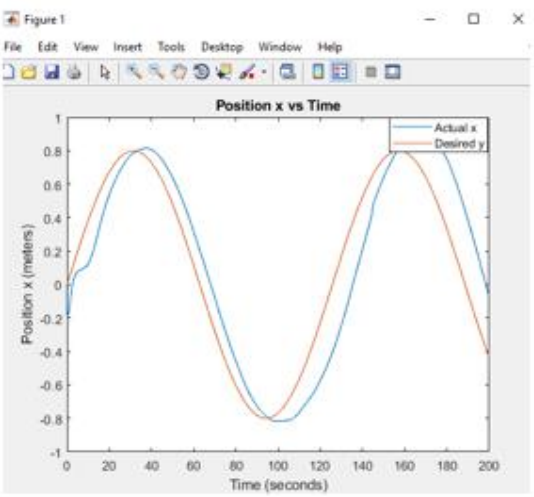


Figure 35 (b)

Figure 35a and Figure 35b: Position x with respect to time with and without accommodation for combinational sensor and LOE faults in both wheels of the robot.

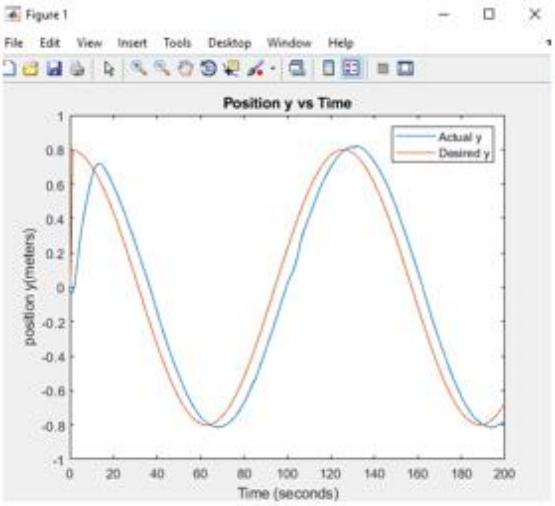


Figure 36 (a)

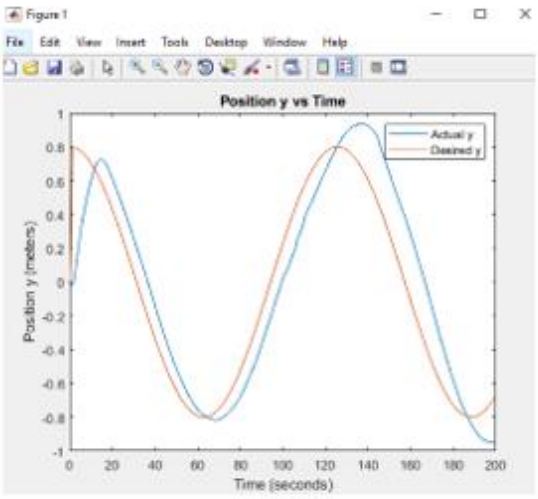


Figure 36 (b)

Figure 36a and Figure 36b: Position y with respect to time with and without accommodation for combinational sensor and LOE faults in both wheels of the robot.

Figure 34 compares the fault-tolerant control tracking performance with the case without fault accommodation. The red line in the plot shows the desired trajectory, and blue line

shows the actual trajectory. These signals are plotted for a time interval of *t=200 seconds.* As can be seen from Figure 34(b), the robot's tracking performance degrades as a result of the faults. In contrast, it is observed from Figure 34(a) that the tracking performance is successfully recovered, illustrating the effectiveness of the adaptive fault accommodation method.

Figure 35 shows the comparison between x axis plots with respect to time for cases with and without accommodation. The red line in the plot shows the desired trajectory, and blue line shows the actual trajectory. As it can be seen from Figure 35(b), The trajectory without accommodation shows fluctuations in the actual path due to the presence of fault in the encoder of the robot. In contrast from Figure 35(a) the tracking performance is smooth proving the effectiveness of the adaptive algorithm. Similarly, from Figure 36 shows comparison of fault-tolerant control performance taking y axis with respect to time to shows the effectiveness of the control algorithm.

5.3.5 Simultaneous faults in two encoders and two motors

1. Combinations of two actuator and two sensor faults (b_r=0.09 and α_r=-0.6, b_l=-0.07 and α_l=-0.5)

This is the case when both of the two motors and encoder sensors of the robot are subjected to simultaneous actuator (LOE) and sensor faults. Specifically, the faults are injected at *t=50 seconds* to right wheel encoder and right wheel motor of the robot with a value of 0.09 and -0.6, respectively. Also, faults are injected at *t=100 seconds* to left wheel encoder and motor with a value of -0.07 and -0.5, respectively.

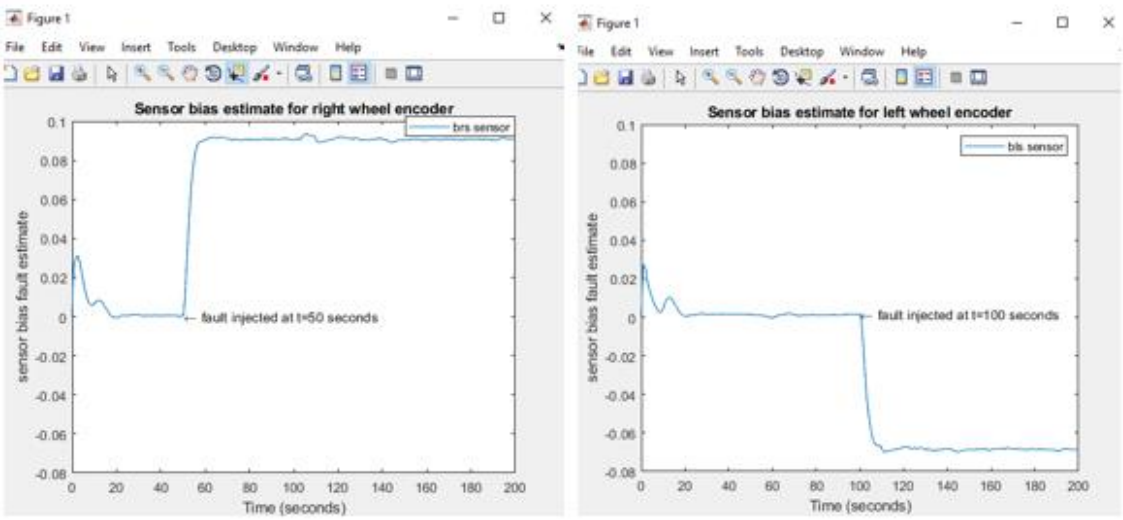


Figure 37 (a) **Figure 37 (b)**

Figure 37a and Figure 37b: Adaptive sensor bias estimates for simultaneous faults in two motors and two encoders.

Figure 37 shows the estimated sensor biases generated by the adaptive algorithm. It can be seen that the estimated sensor biases for both the right wheel and left wheel encoders converges to a value of 0.09 and -0.07, respectively. Additionally, the actuator LOE faults parameter estimates are given in Figure 38. As we can see, the estimated LOE faults for both the right and left wheel motors converge to a value of -0.6 and -0.5, respectively. Thus, the LOE actuator and sensor faults can be successfully identified.

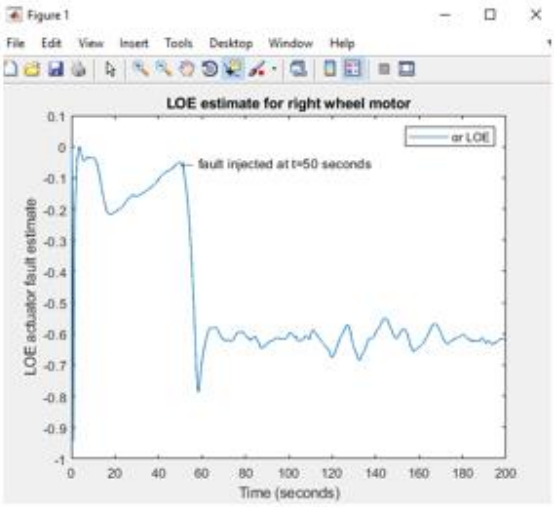


Figure 38 (a)

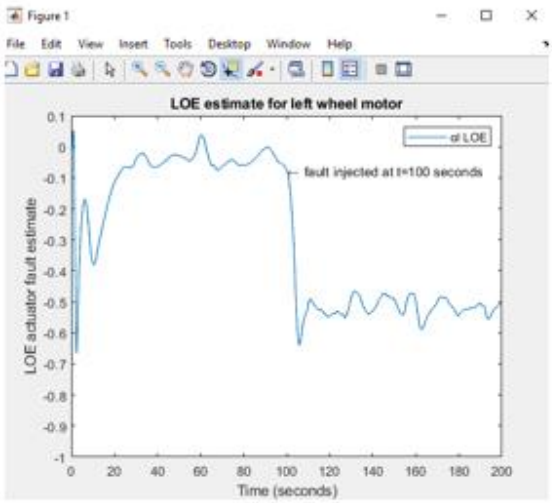


Figure 38 (b)

Figure 38a and Figure 38b: Adaptive actuator LOE fault parameter estimates for simultaneous faults in both encoders and motors.

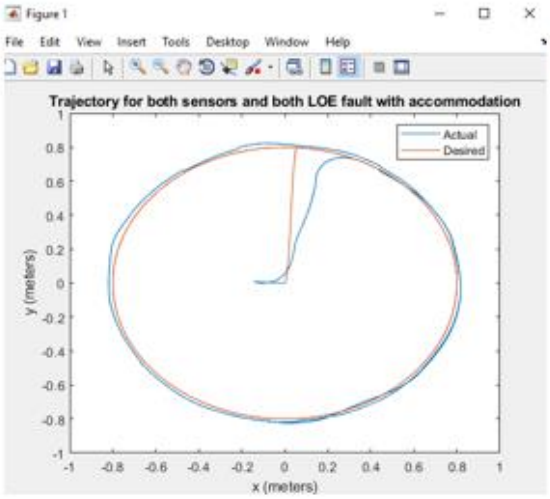


Figure 39 (a)

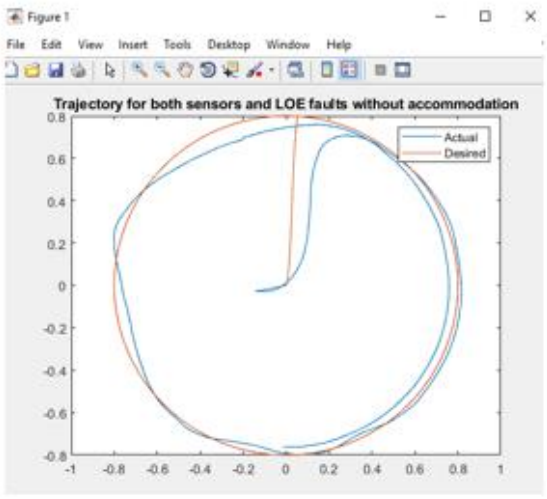


Figure 39 (b)

Figure 39a and Figure 39b: Comparison of tracking performance with and without accommodation.

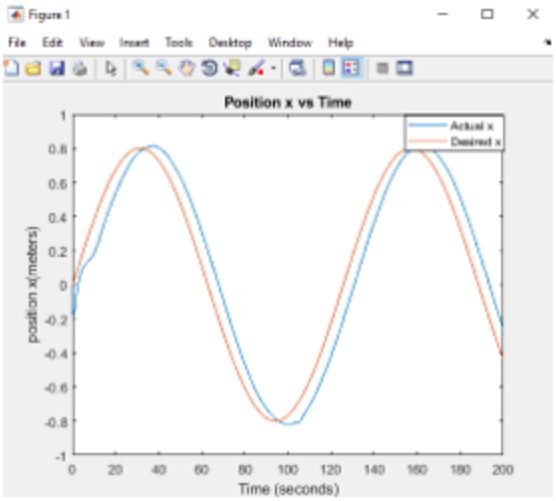


Figure 40 (a)

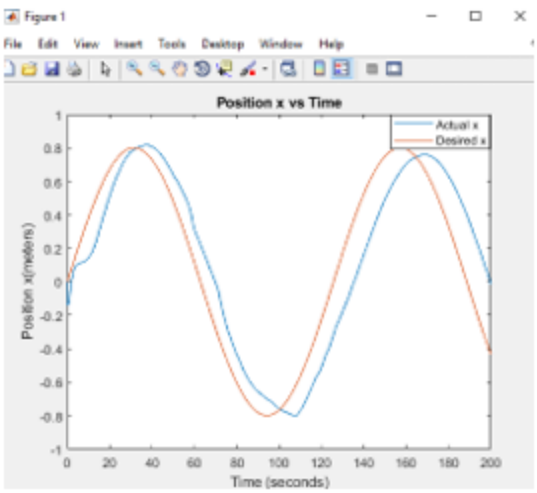


Figure 40 (b)

Figure 40a and figure 40b: Position x with respect to time with and without accommodation for combinational sensor and LOE faults in both encoders and motors of the robot.

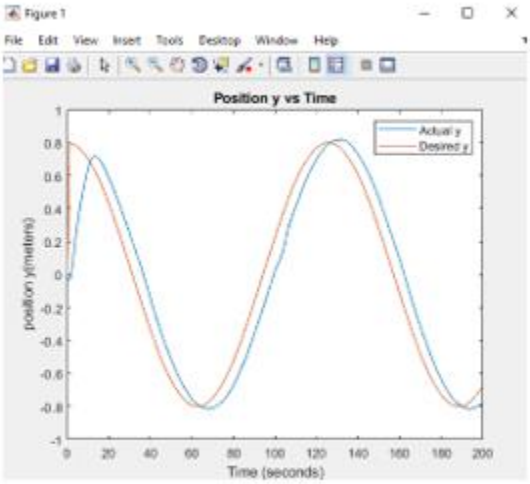


Figure 41 (a)

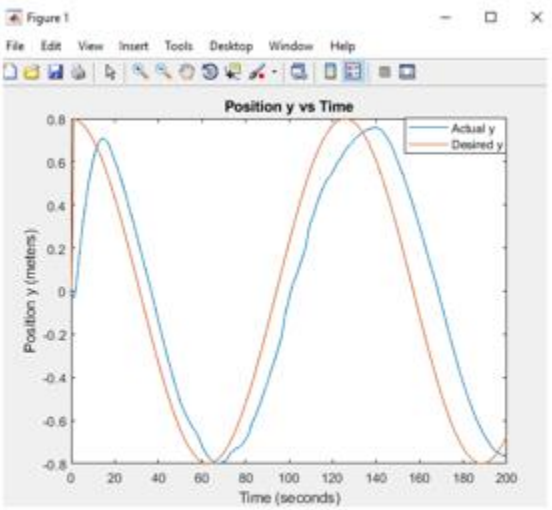


Figure 41 (b)

Figure 41a and figure 41b: Position y with respect to time with and without accommodation for combinational sensor and LOE faults in both encoders and motors of the robot.

Figure 39 compares the fault-tolerant control tracking performance with the case without fault accommodation. The red line in the plot shows the desired trajectory, and blue line

shows the actual trajectory. These signals are plotted for a time interval of *t=200 seconds.* As can be seen from Figure 39(b), the robot's actual trajectory without accommodation clearly deviates from the desired reference trajectory. In contrast, it is observed from Figure 39(a) that the tracking performance is successfully recovered, illustrating the effectiveness of the adaptive fault accommodation method.

Figure 40 shows the comparison between x axis plots with respect to time for cases with and without accommodation. The red line in the plot shows the desired trajectory, and blue line shows the actual trajectory. As it can be seen from Figure 40(b), The trajectory without accommodation shows fluctuations in the actual path due to the presence of fault in the encoder of the robot. In contrast from Figure 40(a) the tracking performance is smooth proving the effectiveness of the adaptive algorithm. Similarly, from Figure 41 shows comparison of fault-tolerant control performance taking y axis with respect to time to shows the effectiveness of the control algorithm.

2. Combinations of two actuator and two sensor faults (b_r=0.09 and α_r=-0.7, b_l=0.06 and α_l=-0.6)

This is the case when both of the two motors and encoder sensors of the robot are subjected to simultaneous actuator (LOE) and sensor faults. Specifically, the faults are injected at *t=50 seconds* to right wheel encoder and right wheel motor of the robot with a value of 0.09 and -0.7, respectively. Also, faults are injected at *t=100 seconds* to left wheel encoder and motor with a value of 0.06 and -0.6, respectively.

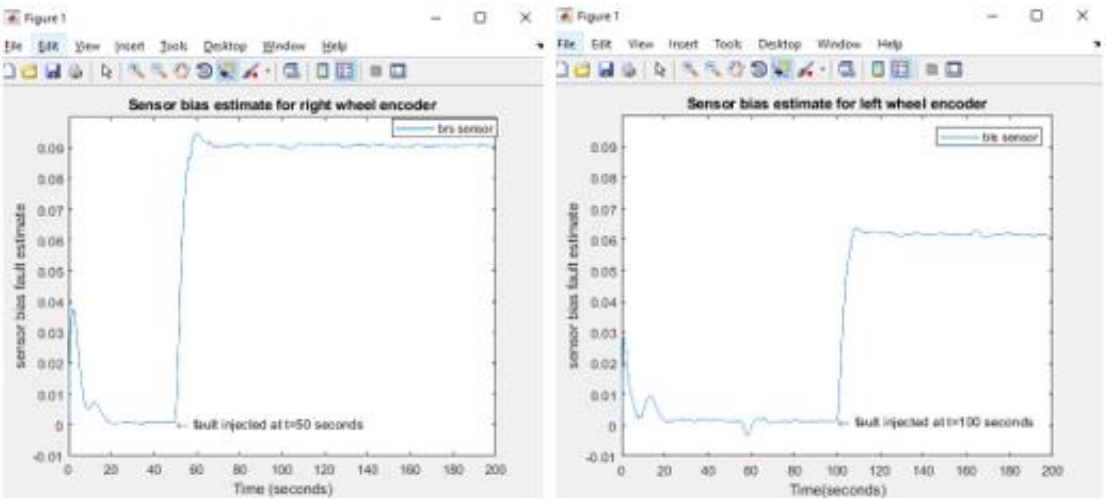


Figure 42 (a)

Figure 42 (b)

Figure 42 and Figure 42b: Adaptive sensor bias estimates for simultaneous faults in two motors and two encoders.

Figure 42 shows the estimated sensor biases generated by the adaptive algorithm. It can be seen that the estimated sensor biases for both the right wheel and left wheel encoders converges to a value of 0.09 and 0.06, respectively. Additionally, the actuator LOE faults parameter estimates are given in Figure 43. As we can see, the estimated LOE faults for both the right and left wheel motors converge to a value of -0.7 and -0.6, respectively. Thus, the LOE actuator and sensor faults can be successfully identified.

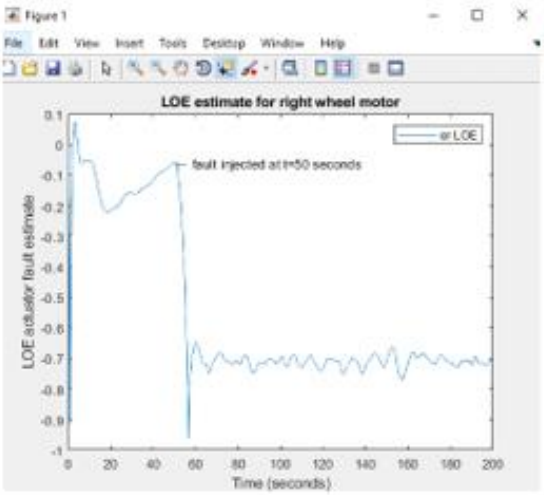


Figure 43(a)

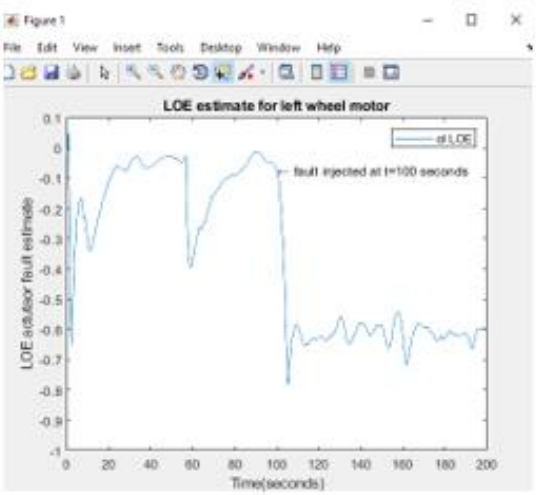


Figure 43 (b)

Figure 43 and Figure 43b: Adaptive actuator LOE fault parameter estimates for simultaneous faults in both encoders and motors.

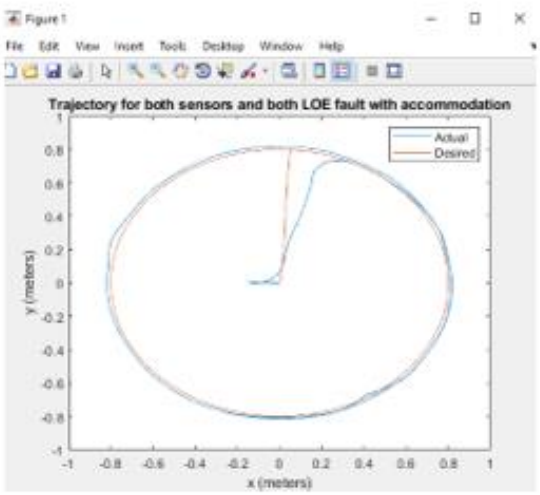


Figure 44 (a)

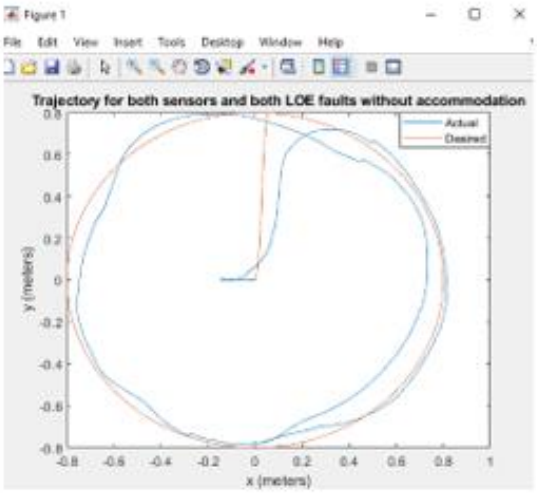


Figure 44 (b)

Figure 44 and Figure 44b: Comparison of tracking performance with and without accommodation.

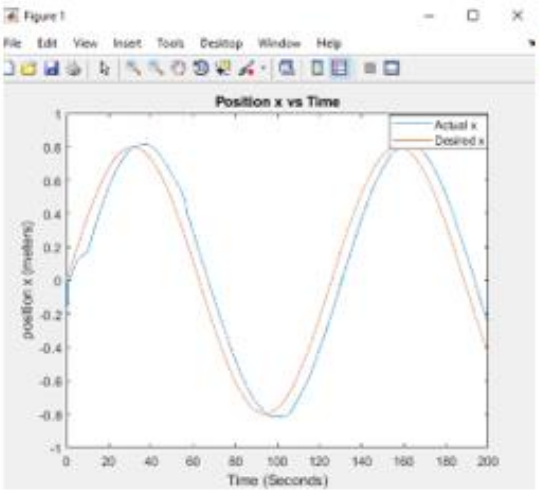


Figure 45 (a)

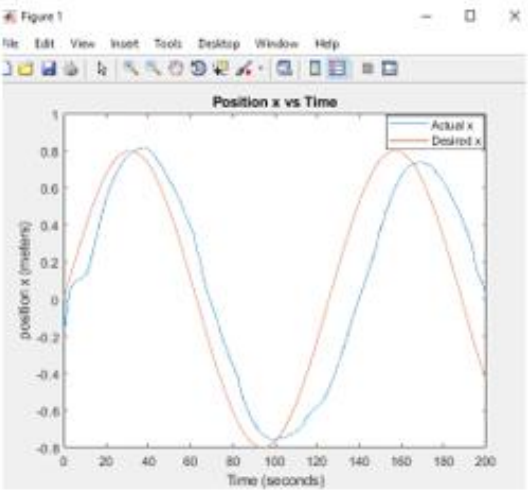


Figure 45 (b)

Figure 45 and figure 45b: Position x with respect to time with and without accommodation for combinational sensor and LOE faults in both encoders and motors of the robot.

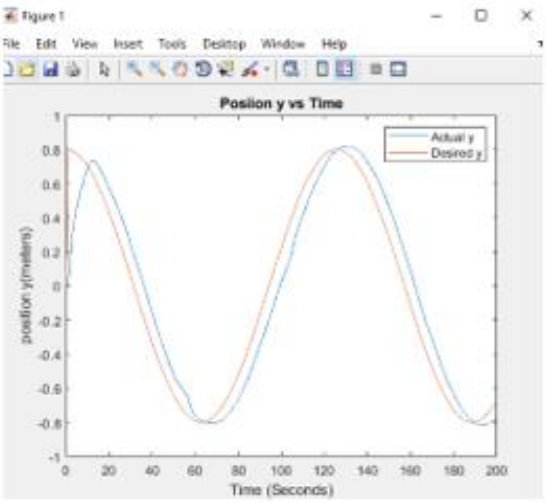


Figure 46 (a)

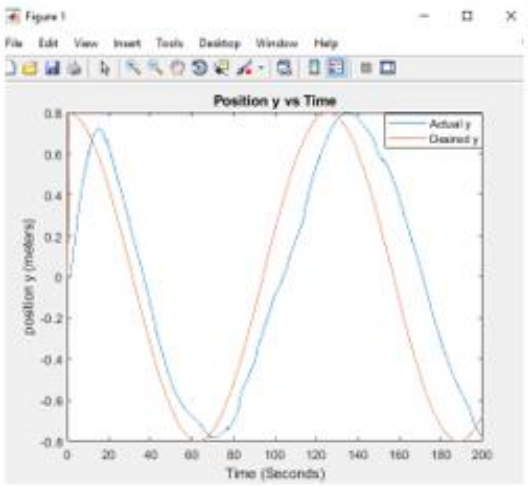


Figure 46(b)

Figure 46 and figure 46b: Position y with respect to time with and without accommodation for combinational sensor and LOE faults in both encoders and motors of the robot.

Figure 44 compares the fault-tolerant control tracking performance with the case without fault accommodation. The red line in the plot shows the desired trajectory, and blue line shows the actual trajectory. These signals are plotted for a time interval of *t=200 seconds.*

As can be seen from Figure 44(b), the robot's actual trajectory without accommodation clearly deviates from the desired reference trajectory. In contrast, it is observed from Figure 44(a) that the tracking performance is successfully recovered, illustrating the effectiveness of the adaptive fault accommodation method.

Figure 45 shows the comparison between x axis plots with respect to time for cases with and without accommodation. The red line in the plot shows the desired trajectory, and blue line shows the actual trajectory. As it can be seen from Figure 45(b), The trajectory without accommodation shows fluctuations in the actual path due to the presence of fault in the encoder of the robot. In contrast from Figure 45(a) the tracking performance is smooth proving the effectiveness of the adaptive algorithm. Similarly, from Figure 46 shows comparison of fault-tolerant control performance taking y axis with respect to time to shows the effectiveness of the control algorithm.

6 CONCLUSION AND FUTURE RESEARCH

6.1 Conclusion

The purpose of this research is to develop and implement adaptive fault estimation and accommodation methods for actuator and sensor faults in ground robot. Various real-time experiments are conducted to verify the effectiveness of the fault-tolerant control algorithm.

The important contribution of this research is the development of a fault diagnosis and fault-tolerant control algorithm to identify and accommodate faults in the ground vehicle and maintain satisfactory tracking performance even in the presence of faults. Actuator faults (LOE) and sensor bias faults were considered. The algorithm was able to successfully identify the fault type, estimate the fault parameters and accommodate the fault effects.

Fault accommodation is conducted by taking advantage of the fault information obtained during the adaptive estimation stage, hence achieving a seamless integration of fault diagnosis and accommodation tasks.

6.2 Future Research

In this research, the robot pose measurements are provided by the Vicon system, which clearly is limited by its coverage area. An interesting extension is to replace the Vicon measurements with a SLAM system.

SLAM is an abbreviation for Simultaneous Localization and Mapping which can be implemented with ROS (Robot Operating System) to provide the robot with real-time pose measurements solely using on-board sensors. Moreover, additional functionalities

including path planning, obstacle avoidance and navigation can be enabled by SLAM, making the robot more robust and versatile with its capabilities.

Another interesting direction for future research is to investigate the application of artificial intelligence methods to the fault-tolerant control problem under consideration. For instance, reinforcement learning method has shown great potential for complicated control tasks. It would also be interesting to compare the performance of reinforcement learning with the adaptive fault-tolerant control method presented in this research.

7 REFERENCES

[1] P. A. Ioannou and J. Sun, *Robust Adaptive Controls*. Mineola, NY, USA: Dover, 1996.

[2] P.A. Ioannou and B, Fidan, *Adaptive Control Tutorial*, SIAM, 2006

[3] J.A. Farrell and M.M. Polycarpou, *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches*, Wiley, April 2006.

[4] R. C. Avram. X. Zhang, and J. Muse, "Quadrotor actuator fault diagnosis and accommodation using nonlinear adaptive estimators," IEEE Trans. Control Syst. Technol., published online, doi: 10.1109/TCST.2016.2640941, Jan. 2017.

[5] Quanser Student Workbook QBot2e for QUARC .

[6] D. Zhuo-hua, C. Zhi-xing, Y. Jin- xia, "Fault Diagnosis and Fault-Tolerant Control for Wheeled Mobile Robots under Unknown Environments: A Survey", Proceedings of the 2005 IEEE Conference on Robotics and Automation, Barcelona, Spain, April 2005.

[7] A. Carvalho Leite, B. Schafer, and M. Lopes de Oliveira e Souza. "Fault-Tolerant Control Strategy for Steering Failure in Wheeled Planetary Rovers", vol. 2012, pp. 1-15.

[8] C. Wu, R. Sehab, A. Akrad, C. Morel, "Fault Diagnosis Methods and Fault-Tolerant Control Strategies for the Electrical Vehicles Powertrains", Energies 15, no. 13, 2022.

[9] P. Guo, H. Kim, N. Virani, Jun Xu, M. Zhu and P. Liu, "Exploiting Physical Dynamics to Detect Actuator and Sensor Attacks in Mobile Robots", arXiv preprint, arXiv:1708.01834, 2017.

[10] J. Zhang, G. Rizzoni, A. Cordoba- Arenas, A. Amodio, “Model- based diagnosis and fault-Tolerant control for ensuring torque functional safety of pedal-by-wire systems, vol. 61, pp. 255-269, April 2017.

[11] P. Goel, G. Dedeoglu, S. I. Roumeliotis and G. S. Sukhatme, "Fault detection and identification in a mobile robot using multiple model estimation and neural network," Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), vol.3, pp. 2302-2309, 2000.

[12] F. Alyoussef, A. Akrad, R. Sehab, C. Morel, and I. Kaya, “Velocity Sensor Fault-Tolerant Controller for Induction Machine Using Intelligent Voting Algorithm”, Energies 15, no. 9, 2022.

[13] A. L. Christensen, R. O’Grady, M. Birattari, M. Dorigo, “Fault detection in autonomous robots based on fault injection and learning”, Autonomous Robot 24, pp. 49–67, 2008.

[14] M. Kinnaert, “Fault Diagnosis based in Analytical Model for Linear and Nonlinear Systems- A Tutorial”, IFAC proceedings, vol. 36, pp. 37-50, June 2003.

[15] G. Zhang, H. Zhang, X. Huang, J. Wang, H. Yu, R. Graaf, “Active Fault-Tolerant Control for Electric Vehicles with Independently Driven Rear In-Wheel Motors Against Certain Actuator Faults”, IEEE Transaction on Control System Technology, vol. 24, no. 5, September 2016.

[16] M. A. Kamel, Y. Zhang, X. Yu, “Fault-Tolerant Cooperative Control of Multiple Wheeled Mobile Robots Under Actuator Faults, IFAC proceedings, vol. 48, no. 21, pp. 1152–1157, 2015.

[17] Y. Wang, S. Yu, J. Yuan, and H. Chen "Fault-Tolerant Control of Electric Ground Vehicles Using a Triple-Step Nonlinear Approach, IEEE/ASME Transaction on Mechatronics, vol. 23, no. 4, pp. 1775-1786, August 2018

[18] E. Khalastchi, M. Kalech," Fault Detection and Diagnosis in Multi-Robot Systems: A Survey", Vol. 19(18), September 2019

[19] X. Fu, Y. Yuan*, D. Liu, Q. Zhang, J. Pei, "Fault Tolerance Control of Four In-wheel Motors Driven Vehicles Based on Limping Dynamic Performance", 2020 3rd International Conference on Electron Device and Mechanical Engineering (ICEDME), pp. 352-359, 2020.

[20] S. Lakhera, A. Mishra, S. Sengupta, "Fault Tolerant Control of Electric Vehicles with Transmission Fault", 2019 IEEE 5th International Conference for Convergence in Technology (I2CT) , pp. 1-7, 2019.

[21] H. Dai, P. Chen and H. Yang, "Metalearning-Based Fault-Tolerant Control for Skid Steering Vehicles under Actuator Fault Conditions", Sensors 22, no. 3:845, published online, doi.org/10.3390/s22030845, 2022.

[22] Bellur, Ravindra Vibha, "Real-time Path Planning and Obstacle Avoidance for Mobile Robots with Actuator Faults", Wright State University, 2018.