

Article

Not peer-reviewed version

Fault Detection of Multi Wheeled Robot Consensus Based on EKF

[AFRAH JOUILI](#)*, [Boumedyen Boussaid](#)*, [Ahmed Zouinkhi](#), Naceur. Abdelkrim

Posted Date: 9 May 2024

doi: 10.20944/preprints202405.0585.v1

Keywords: Wheeled Mobile Robot; master-slave system; Non-linear system; Fault detection and 9 isolation (FDI); residues; Extended Kalman Filter



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Fault Detection of Multi Wheeled Robot Consensus Based on EKF

Afrah JOUILI ^{1,†}, Boumedyen BOUSSAID ^{2,*,†,‡}, Ahmed ZOUINKHI ^{3,†} and M.N ABDELKRIM ^{4,†}

¹ afrah.jouili.enig@gmail.com

² boumedyen.boussaid@eseo.fr

³ ahmed.zouinkhi@enig.rnu.tn

⁴ naceur.abdelkrim@enig.rnu.tn

* Correspondence: boumedyen.boussaid@eseo.fr

† MACS Research Laboratory, Engineering School of Gabes, University of Gabes, 6029, Zrig, Tunisia

‡ ESEO, Western Higher School of Electronics, 78140 Vélizy-Villacoublay, France

Abstract: In this article the problem of detection and isolation (FDI) in wheeled mobile robot is treated. first of all we consider a master-slave robot system in which the master robot has to follow a desired trajectory given by a well-determined control law, then the slave is controlled by the master. However in real environment there is no ideality because faults can appear in the master or slave or even the both and affect the system and may lead to unacceptable results so to ensure a well functioning an extended kalman filter is tuned to diagnosis eventual faults and to generate residuals in order to detect and isolate them. The performance of an estimator based on an EKF is simulated based MATLAB Simulink simulations.

Keywords: wheeled mobile robot; master-slave system; non-linear system; fault detection and isolation (FDI); residues; extended kalman filter

0. Introduction

In recent years, there has been a significant shift in focus towards cooperative multi-agent control (MAS) within the realm of control research. This shift is driven by its rapid advancement across various domains, including several robot systems, intelligent transportation, and numerous industrial setups [1–5]. The primary benefit of MAS lies in its ability to enhance system functionality, often overcoming challenges that may prove difficult for humans or even entirely superseding human involvement in executing tasks that are repetitive, hazardous, or beyond human capability [6]. Consequently, the importance of fault detection and isolation has escalated, ensuring safety and maintaining quality standards. Given the inevitability of faults in embedded systems such as wheeled mobile robots (WMRs), it is imperative to ensure they are promptly identified and addressed.

Within computer networking, the master-slave model serves as a communication protocol wherein a designated device or process (referred to as the master) governs one or more other devices or processes (the slaves). Once the master-slave relationship is established, control flows exclusively from the master to the slave(s).

Master-slave systems offer simplicity and provide assurance for both system integrity and data security. However, their significant drawback arises in the event of master failure, which can lead to the entire system's failure, causing process malfunctions and subsequent degradation in performance [7]. Hence, the necessity of implementing process monitoring and fault detection systems becomes paramount, especially in applications like wheeled robots. This prompts exploration into fault diagnosis techniques to effectively address this challenge. Fault detection and isolation (FDI) has garnered substantial attention in both academic and industrial spheres, as evidenced by recent publications, for example, [8]. FDI constitutes a process facilitating the identification of faulty components. According to [9], this methodology is typically categorized into three overarching groups: history-based methods, hardware-based methods, and model-based methods. Within these classifications, two primary classes emerge: qualitative approaches and quantitative approaches. In literature, considerable attention has been directed towards the observer-based technique. For instance, for example in [10], a method

utilizing Kalman filter identification is utilized for the detection and isolation of sensor faults within mobile robotic systems. Also in [11], presents the integration of a bank of Kalman filters with an expert system for the detection and isolation of sensor faults in mobile robots. Additionally, [12] introduces a Kalman filter designed for joint state prediction and unknown input estimation within linear stochastic discrete-time systems subject to intermittent unknown inputs in measurements. Interests in FDI for nonlinear systems have grown significantly in recent years due to the fact that most of the systems, we face in practice, are nonlinear in nature such as [13] where the FDI system is based on a single model EKF filter that generates residuals as soon as the behavior of the aircraft deviates from expected, also [14] directs its focus towards fault detection in wheeled mobile robots utilizing an EKF filter. The fault detection process typically involves two primary steps: residual generation and subsequent residual evaluation. In our contribution, we will emphasize the observer-based approach, particularly the utilization of the Extended Kalman Filter.

To investigate fault detection and isolation within a nonlinear system, specifically a wheeled mobile robot, utilizing the Extended Kalman Filter, this paper is structured as follows: The first section outlines the modeling of the unicycle mobile robot. In the subsequent section, we delve into the Extended Kalman Filter (EKF) as a corrective predictor technique. This section covers two primary tasks encompassing prediction and correction phases, alongside the fault diagnosis steps which include residual generation, evaluation, and decision-making processes.

Following this, simulations are presented to illustrate the behaviors of both master-robot and slave-robot, showcasing how the EKF detects faults and generates residuals. Finally, the last section concludes the paper.

1. Multi-Wheeled Robot Consensus

1.1. Graph Theory

1.1.1. Notation

We denote the set of real numbers as $\mathbb{R}$. The notation $\mathbb{R}^n$ represents the real vector space with n dimensions, while $\mathbb{R}^{n \times n}$ denotes the set of $n \times n$ matrices. A diagonal matrix of size $n \times n$, with diagonal elements $q_1, q_2, \dots, q_n$, is denoted as $\text{diag}(q_1, q_2, \dots, q_n)$. The identity matrix is represented by $I \in \mathbb{R}^{n \times n}$. The Kronecker product of matrices is denoted by $\otimes$. Additionally, we utilize $x = (x_1, \dots, x_n)^T$ to signify the vector in $\mathbb{R}^n$. T in exponential means the transposition.

1.1.2. Graph Theory

Understanding fundamental concepts of graph theory is indispensable for exploring the dynamics of multi-agent systems [15]. Consider a multi-agent system comprising n mobile robots, interconnected through a communication network. The interaction model among agents can be represented by delineating the communication topology in the form of a graph

Consider $\mathcal{G}_t = (\Lambda, Y)$ an oriented graph, where $\Lambda = [\Lambda_1, \Lambda_2, \dots, \Lambda_n]$ represents the set of nodes (vertices), Each node i signifies the i^{th} agent, and

$$Y = (i, j) \in \Lambda \times \Lambda, j \in N_i$$

represents the collection of edges (arrows) linking the nodes, where N_i denotes the set of node indexes connected to the i^{th} node. In the notation (i, j) , it signifies a connection between nodes i and j , with j being in N_i , although i may not be in N_j . Therefore, (i, j) and (j, i) do not necessarily denote the same edge. These node connections are consolidated into a matrix $M_d = [a_{ij}]$, known as the adjacency

matrix or connection matrix, with dimensions $N * N$. The elements within the matrix indicate whether pairs of vertices are adjacent in the graph or not.

$$a_{kj} = \begin{cases} 1 & \text{if } k \text{ is connect to } j \\ 0 & \text{if not.} \end{cases}$$

The degree matrix Q is a diagonal matrix of size $N \times N$, providing information regarding the number of edges connected to each node. With

$$Q_{ij} = \begin{cases} \text{degree}(\Delta_i) & \text{if } i = j \\ 0 & \text{if not.} \end{cases}$$

We define the Laplacian matrix $H = [l_{ij}]$ by:

$$H = Q - M$$

The components within this matrix are as follows::

$$h_{kj} = \begin{cases} -a_{kj} & \text{if } k \neq j \\ \sum_{j=1}^N a_{kj} & \text{if } k = j. \end{cases}$$

1.1.3. Closed Loop System

Consider a collection of $N = 3$ agents of significant scale, wherein the conduct of each agent is delineated by a non-linear controlled model.

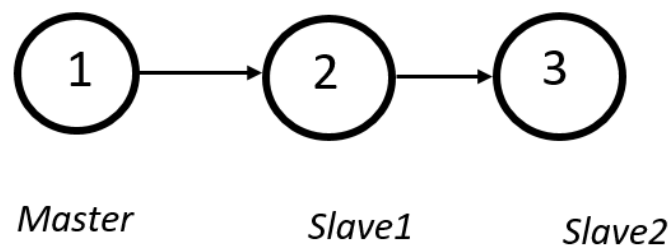


Figure 1. Directed graph.

The Figure 1 shows the master-slave configuration considered in this paper so the matrices are as follow.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } L = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

Remember also the writing of the protocol:

$$u_i = -P(X_i) \left(\sum_{j=1}^N a_{ij} (X_i - X_j) \right) \quad (1)$$

which $P(X_i) \in \mathbb{R}^{n \times m}$, a_{ij} are the adjacent elements linked to $\mathcal{G}$

1.2. Wheeled Mobile Robot Modeling

An unicycle is a robot powered by two independent wheels. It is placed on a plane surface where $(0, \vec{x}, \vec{y})$ an inertial reference. The kinematic model illustrated in Figure 2 and the continuous state-space representation is given as follows:

$$\begin{cases} \dot{x}(t) = v \cos(\theta(t)) \\ \dot{y}(t) = v \sin(\theta(t)) \\ \dot{\theta}(t) = \omega(t) \\ \dot{v}(t) = a(t) \end{cases} \quad (2)$$

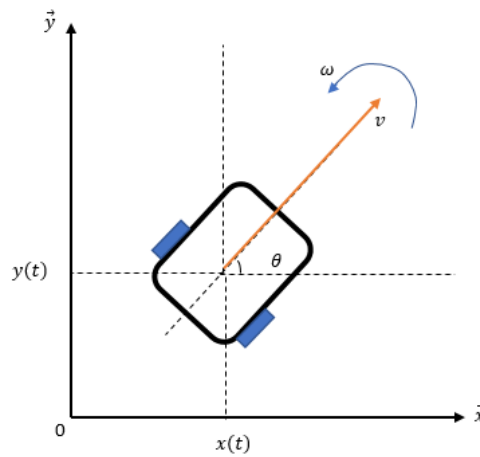


Figure 2. Geometry of an unicycle robot

Where x and y are the position along the x -axes and the y -axes respectively, $v \in \mathbb{R}$ denotes the linear speed associated with the center of mass, θ is the orientation angle, ω is the angular speed of the center of mass, a is the acceleration and u represents the input vector $u = \begin{bmatrix} \omega & a \end{bmatrix}^T$. Then, the generalised robot model is given by:

$$\begin{cases} \dot{X}(t) = f(X, u, t) \\ Y(t) = CX(t) \end{cases} \quad (3)$$

With $X = \begin{bmatrix} x & y & \theta & v \end{bmatrix}^T$ and The observation matrix $C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ it provides the system observability condition.

We propose to calculate a regulator which makes it possible to describe an equation of a sinusoidal trajectory.

$$\begin{aligned} x_d(t) &= t \\ y_d(t) &= \sin(t) \end{aligned} \quad (4)$$

For this, we use a method by linearizing looping. The command will therefore be in the end as follow:

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -v \sin(\theta) & \cos(\theta) \\ v \cos(\theta) & \sin(\theta) \end{pmatrix}^{-1} \begin{pmatrix} (x_d - x) + 2(\dot{x} - v \cos(\theta)) + \ddot{x}_d \\ (y_d - y) + 2(\dot{y} - v \sin(\theta)) + \ddot{y}_d \end{pmatrix} \quad (5)$$

If we define the error vector,

$$e = (e_x, e_y)$$

the dynamics of the error is written as follow:

$$\begin{pmatrix} e_x + 2\dot{e}_x + \ddot{e}_x \\ e_y + 2\dot{e}_y + \ddot{e}_y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

which is stable and converges rapidly to 0. According to Euler's approximation, we will discretize the state model (1) by a sampling time T_e as follows:

$$\begin{cases} x_{k+1} = x_k + T_e v_k \cos(\theta_k) \\ y_{k+1} = y_k + T_e v_k \sin(\theta_k) \\ \theta_{k+1} = \theta_k + T_e \omega_k \\ v_{k+1} = v_k + T_e a_k \end{cases} \quad (6)$$

1.3. Simulations

NOTATION:

m : The robot-master

$s1$: The first robot-slave

$s2$: The second robot-slave

$X_d = \begin{bmatrix} x_d & y_d & \theta_d & v_d \end{bmatrix}^T$: State vector of the desired trajectory

$X_m = \begin{bmatrix} x_m & y_m & \theta_m & v_m \end{bmatrix}^T$: State vector of the master robot

$X_{s1} = \begin{bmatrix} x_{s1} & y_{s1} & \theta_{s1} & v_{s1} \end{bmatrix}^T$: State vector of the first slave robot

$X_{s2} = \begin{bmatrix} x_{s2} & y_{s2} & \theta_{s2} & v_{s2} \end{bmatrix}^T$: State vector of the second slave robot

$u_m = \begin{bmatrix} u_{ml} & u_{mr} \end{bmatrix}^T$: The left and right controller of the robot-master

$u_{s1} = \begin{bmatrix} u_{s1l} & u_{s1r} \end{bmatrix}^T$: The left and right controller of the robot-slave

$u_{s2} = \begin{bmatrix} u_{s2l} & u_{s2r} \end{bmatrix}^T$: The left and right controller of the robot-slave

Consider the desired trajectory in Figure 9.

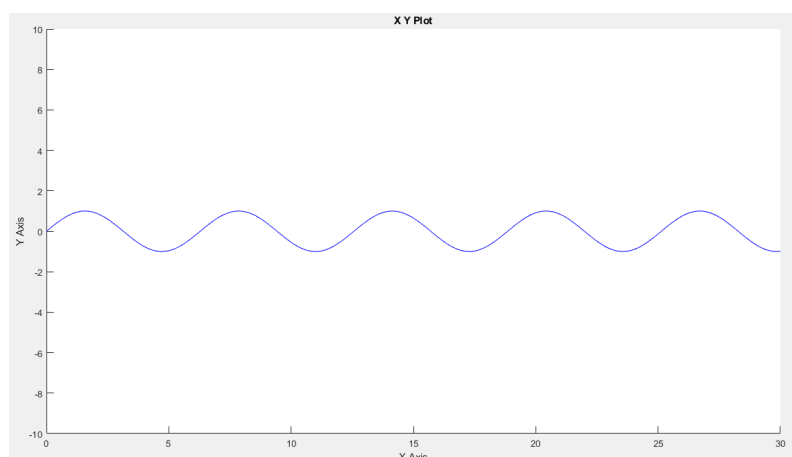


Figure 3. The desired trajectory.

We observe that the robot-master accurately follows this trajectory, which is also replicated by the two robot slaves, as depicted in Figures 4–6.

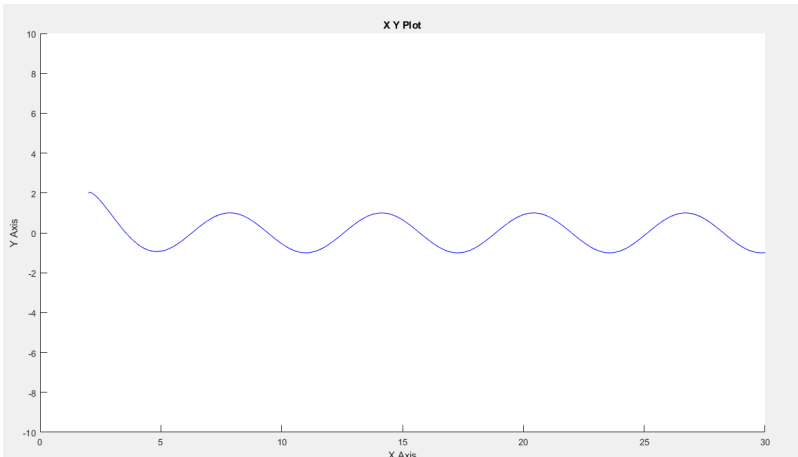


Figure 4. The robot-master’s trajectory

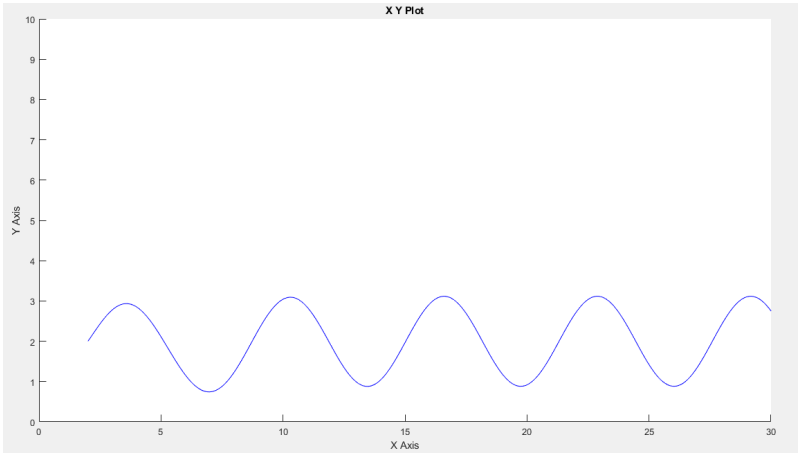


Figure 5. The first robot-slave’s trajectory

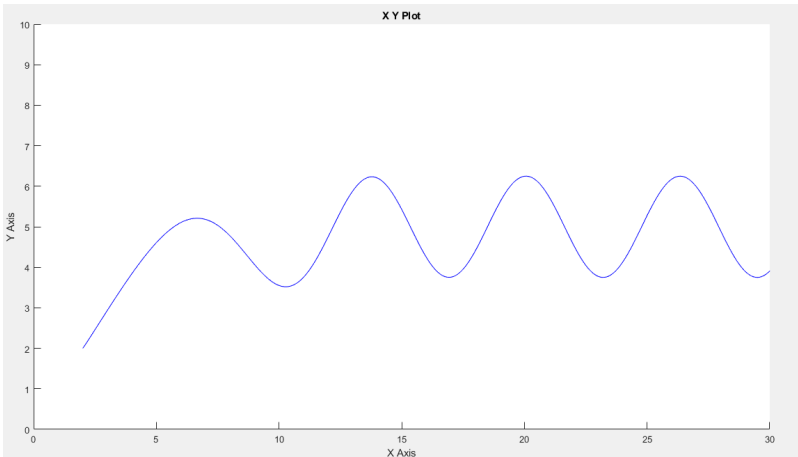


Figure 6. The second robot-slave’s trajectory

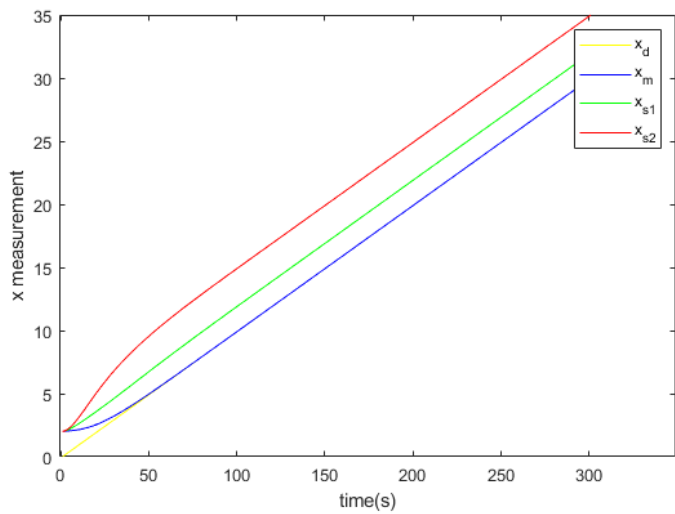


Figure 7. The coordinate x of the desired trajectory, the master and the slaves

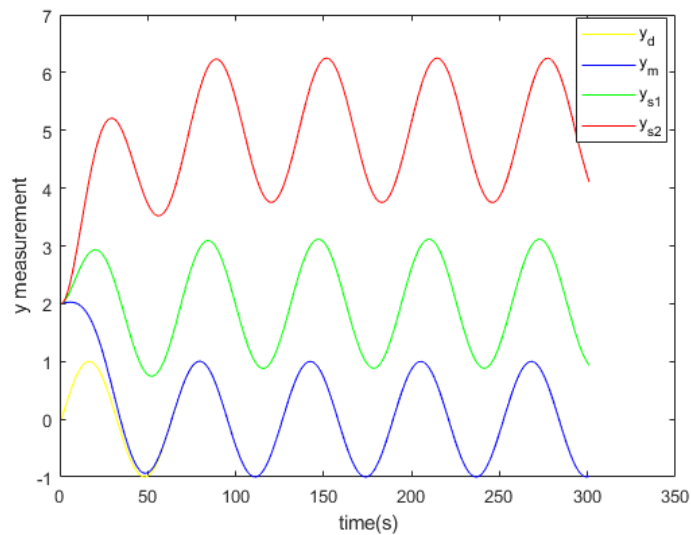


Figure 8. The coordinate y of the desired trajectory, the master and the slaves

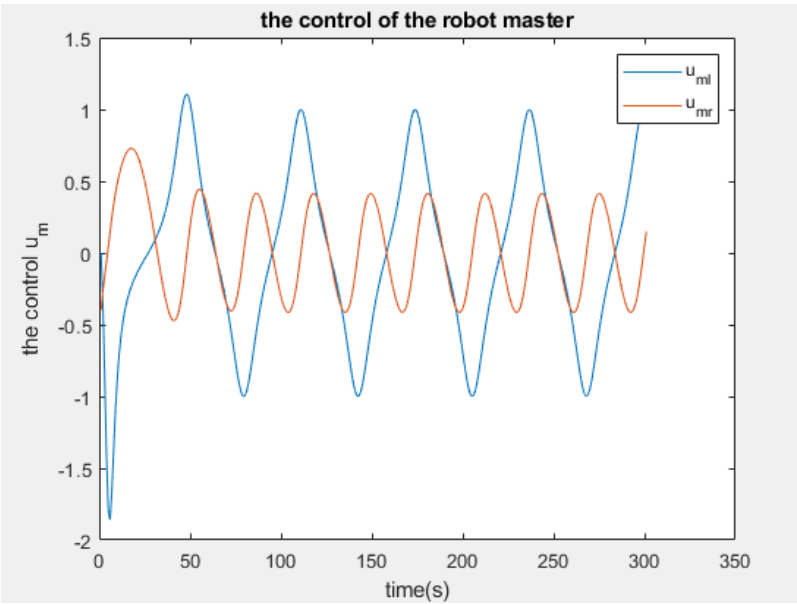


Figure 9. The control of the robot-master

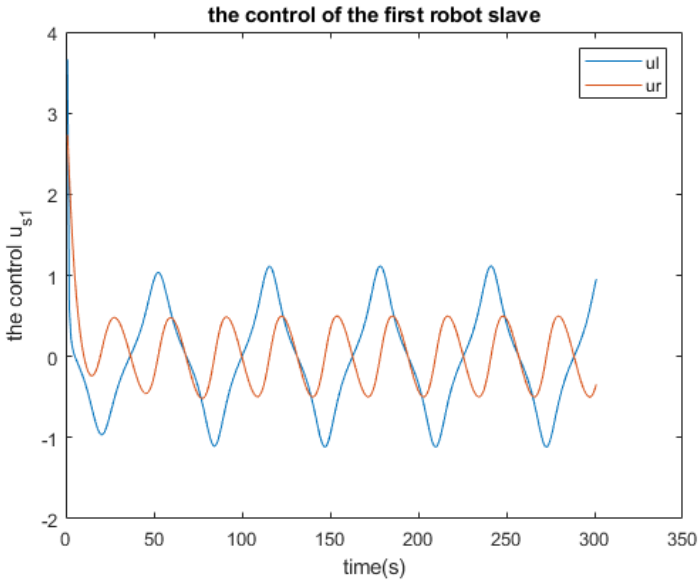


Figure 10. The control of the first robot-slave

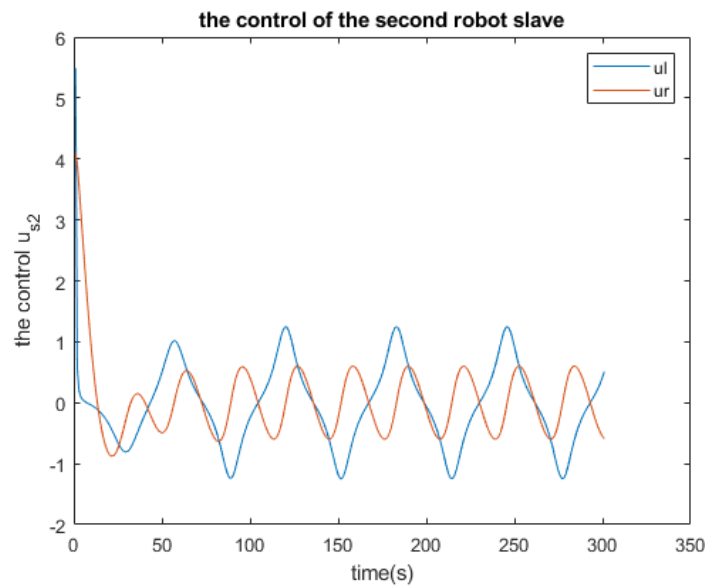


Figure 11. The control of the second robot-slave

2. Fault Detection Based on the Extended Kalman Filter

2.1. Extended Kalman Filter and Fault Diagnosis

2.1.1. the Extended Kalman Filter (EKF)

Within the extended Kalman filter framework, the nonlinearities inherent in the system dynamics are approximated using a linearized rendition of the nonlinear model based on the most recent state estimate. Let's consider the system characterized by the following state equations:

$$\begin{cases} X(k+1) = f(X, u, k) + \rho(k) \\ Y(k+1) = h(X, u, k) + \gamma(k) \end{cases} \quad (7)$$

$\rho(k)$ and $\gamma(k)$ denote the process and observation noises, assumed to have zero mean with covariances Ψ_{ρ_k} and Ψ_{γ_k} respectively. Function f , defined as non-linear according to Equation (7), is employed to compute the predicted state from the previous estimate, as is function h for predicting the measurement from the predicted state. However, due to the nonlinearity of f and h , a Jacobian matrix of partial derivatives is computed since these functions cannot be directly applied to the covariance.

Hypothesis:

As stated in [16], when the Extended Kalman Filter (EKF) is applied to a system and placed in a canonical observation form, it gains properties associated with global convergence.

- The pair of the matrices (A, C) is detectable, that means there is no unstable mode and unobservable in the system
- the signals $\rho(k)$ and $\gamma(k)$ are central Gaussian white noise of Density Power Spectral (DSP) covariance Ψ_{ρ_k} and Ψ_{γ_k} respectively means

$$\begin{cases} \mathbb{E}[\gamma(i)\gamma(j)^T] = \Psi_{\gamma_k}, & \text{if } i = j \\ \mathbb{E}[\rho(i)\rho(i)^T] = \Psi_{\rho_k}, & \text{if } i = j \\ \mathbb{E}[\gamma(i)\rho(j)^T] = 0, & \forall i, j. \end{cases} \quad (8)$$

Where $\mathbb{E}[\cdot]$ represents the mathematical expectation. The latter equation illustrates the stochastic independence of the noises ρ and γ . This assumption is introduced to simplify the subsequent

calculations but is not obligatory.

Correction:

- Updated state estimate :

$$\hat{x}_{k/k} = \hat{x}_{k/k-1} + \mathbb{K}_k \hat{y}_k \quad (9)$$

- Updated covariance estimate :

$$\Psi_{k/k} = (I - \mathbb{K}_k C_k) \Psi_{k/k-1} \quad (10)$$

- measurement residual :

$$\tilde{y}_k = y_k - C_k \hat{x}_{k/k-1} \quad (11)$$

- Innovation covariance:

$$\mathbb{Z}_k = C_k \Psi_{k/k-1} C_k^T + \Psi_{\gamma_k} \quad (12)$$

- Near-optimal Kalman gain:

$$\mathbb{K}_k = \Psi_{k/k-1} C_k^T \mathbb{Z}_k^{-1} \quad (13)$$

Prediction:

- Predicted state estimate :

$$\hat{x}_{k+1/k} = A_k \hat{x}_{k/k} + u_k \quad (14)$$

- Predicted covariance :

$$\Psi_{k+1/k} = A_k \Psi_{k/k} A_k^T + \Psi_{\rho_k} \quad (15)$$

where A_k is the linearized matrix of the function f which is written as follows :

$$A_k = \left. \frac{\delta f}{\delta X} \right|_{\hat{x}_{k/k}}$$

Remark: Occasionally, numerical issues can cause the covariance of innovation Z_k to lose its positive definite nature. In such instances, it is advisable to substitute the equation for covariance estimation with:

$$\Psi_{k/k} = \sqrt{(I - K_k C_k) \Psi_{k/k-1} \Psi_{k/k-1}^T (I - K_k C_k)^T}$$

The replacement will always ensure a positive definite outcome, even if the matrix $\Psi_{k/k-1}$ is not. Consequently, the Kalman equations will exhibit greater stability, as any minor deviations in the positive definite nature of the covariance matrices will be corrected in the subsequent iteration.

According to (14) :

$$A_k = \begin{bmatrix} 1 & 0 & -T_e v \sin(\theta) & \cos(\theta) \\ 0 & 1 & T_e v \cos(\theta) & \sin(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Suppose the signals γ_k and ρ_k are Gaussian white with a unit covariance matrix, that is to say:

$$\Psi_{\gamma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \Psi_{\rho} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{also we take the sample time } T_e = 0.01.$$

2.1.2. Fault Diagnosis Steps

Technically, a fault diagnosis system usually goes through three basic steps: a residual generation, a residue evaluation module, and a decision logic [17].

Step 1: Residual generation

A standard residual generator enables the construction of a fault indicator vector or residuals vector, denoted as r , based on the measured input and output quantities of the system. Under normal conditions without faults in the monitored system, the residual's nominal value theoretically reaches zero once the transient effects from initial conditions have dissipated. However, it deviates notably from zero in the presence of specific faults.

Step 2: Residue evaluation module

The residue evaluation module constitutes the detection phase, where suitable methods are employed to determine whether the system is operating normally or not, i.e., if the given residue deviates from zero [17]. Detecting a fault typically involves comparing the residues to a predetermined detection threshold t_{th} . This threshold represents a crucial aspect of residual-based FDI methods, defined as the boundary value for the deviation of a residue from zero. Thus, the process of fault detection proceeds as follows:

$$r = \begin{cases} 1 & \text{if } r > |t_{th}| \\ 0 & \text{if } r < |t_{th}|. \end{cases}$$

Excessively high thresholds may result in failing to detect a fault (missed alarm), while overly low thresholds may lead to false alarms, detecting faults in healthy conditions [18]. There exist two categories of thresholds [8]: the first being a constant threshold, and the second an adaptive one. The adaptive threshold is employed to accommodate inevitable parameter uncertainty, disturbances, and noise encountered in practical applications.

Step 3: decision

This step involves examining the outcome of evaluating a set of residues and, based on the pattern of activated and non-activated tests, it generates a determination regarding the faulty component within the monitored system.

2.2. Closed Loop System

The closed loop system is given in Simulink/Matlab by the functional diagram in Figure 12.

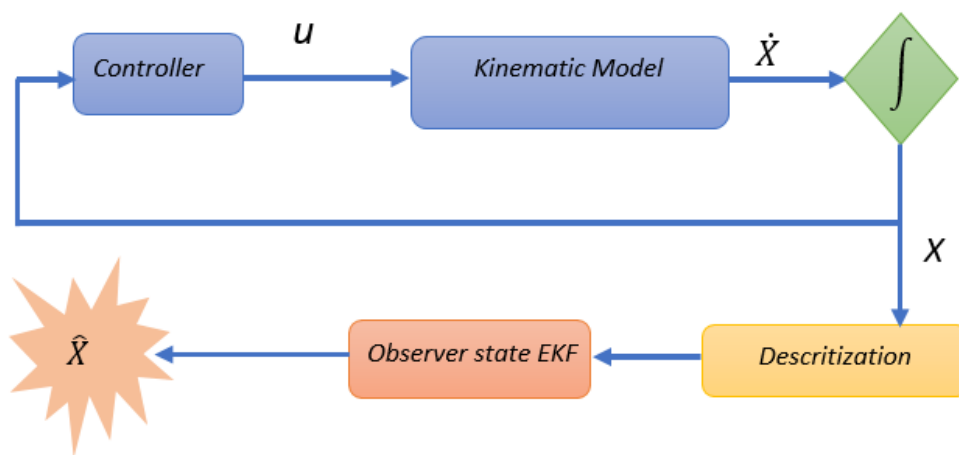


Figure 12. Unicycle mobile robot closed loop

The robot is controlled in order to follow a desired trajectory given by $P_d = [x_d, y_d]^T$ as shown in Figure 13.

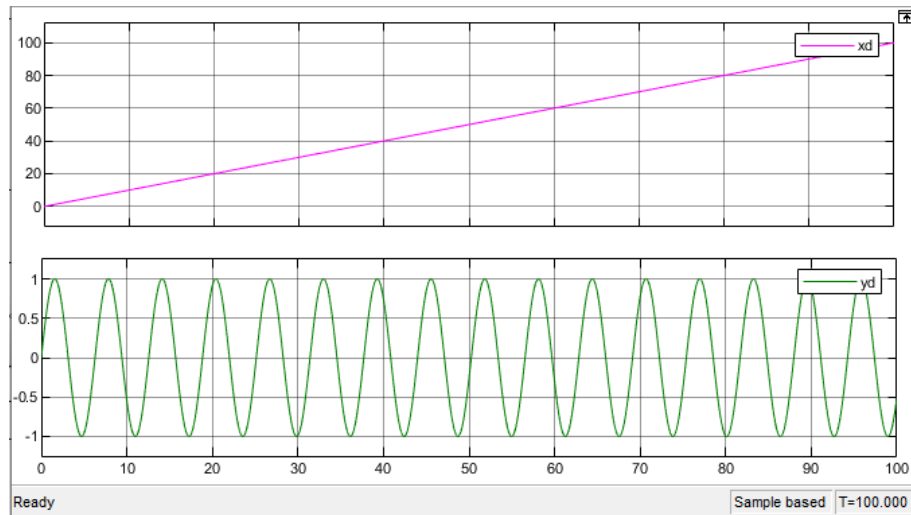


Figure 13. Desired trajectories x_d and y_d

2.3. Simulation Results

First of all, we notice that the robot master follows the desired trajectory, as shown in Figure 14.

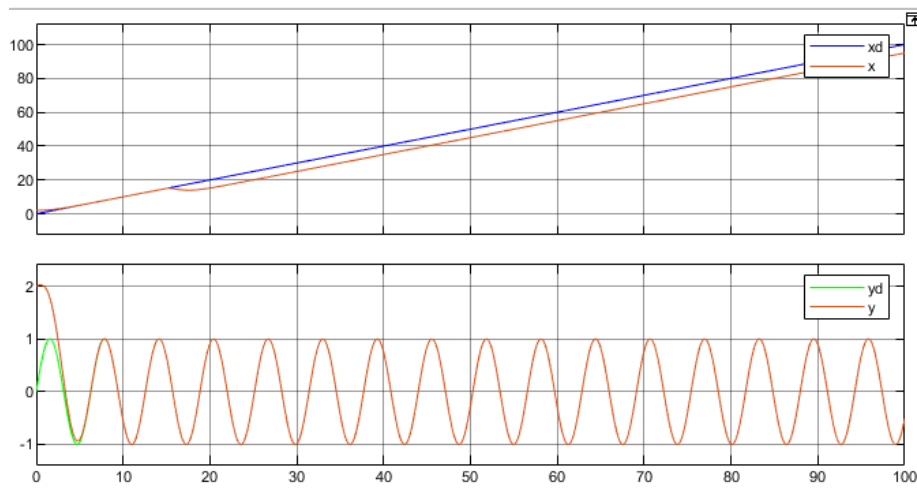


Figure 14. Coordinates x and y of the desired trajectory and the mobile robot

The fault scenario addressed involves an offset applied to the x measurement at the 15-second mark. Subsequently, we apply our Extended Kalman Filter (EKF), and the resulting robot trajectory is illustrated in Figure 15.

It is observed that there is a slight deviation attributable to the applied fault, yet the robot continues to track the same trajectory consistently.

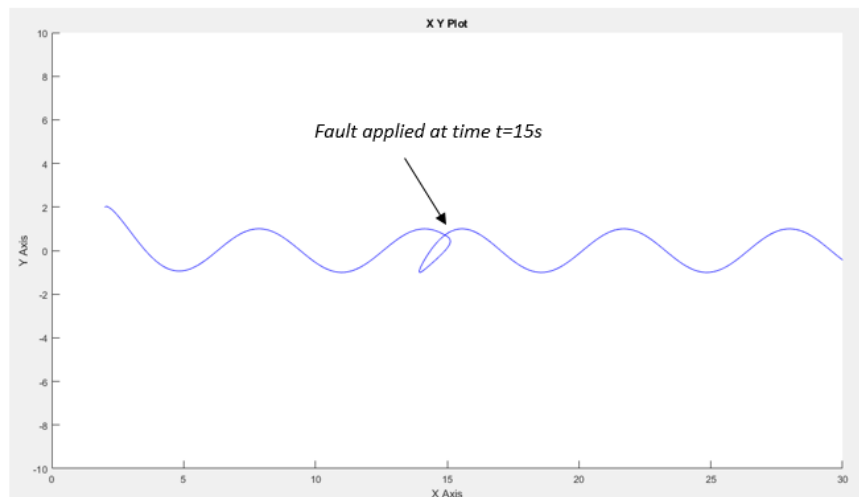


Figure 15. Trajectory of the mobile robot when fault applied at $t=15s$

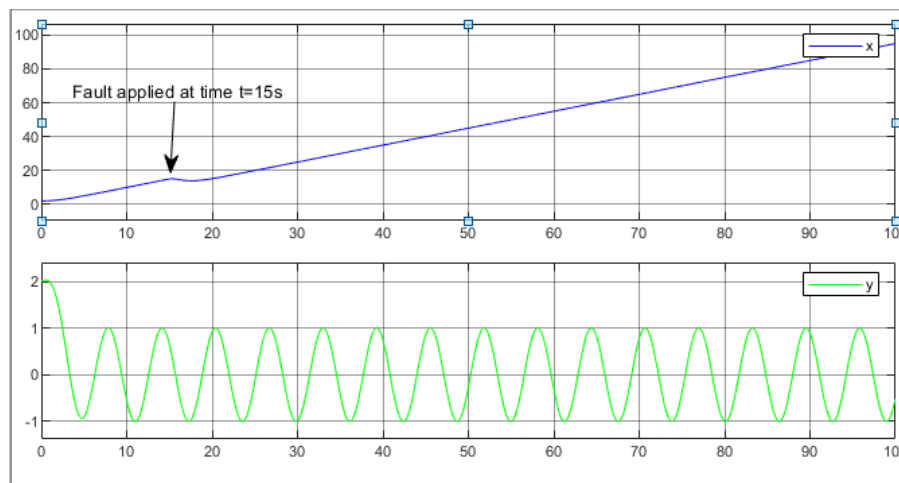


Figure 16. Coordinates x and y of the mobile robot when fault applied at $t=15s$

Furthermore, upon introducing a fault (considered as an intermittent fault, such as a switch) to a parameter at time $t = 15s$, a minor peak is observed, after which the robot promptly returns to its intended trajectory. Hence, it can be concluded that the Extended Kalman Filter effectively identifies the fault introduced into the system.

3. Fault Detection in Multi-Robot System

In the preceding section, the system (consisting of a robot-master and slaves) executes its trajectory within an ideal environment. However, real-world environments lack such ideal conditions. Thus, we aim to introduce a fault and observe how the system responds to the challenge. It's observed that the robot-master adheres to its trajectory, as depicted in Figure 3. Subsequently, we induce an offset fault in the master robot's x measurement at 15 seconds, as illustrated in Figure 18.

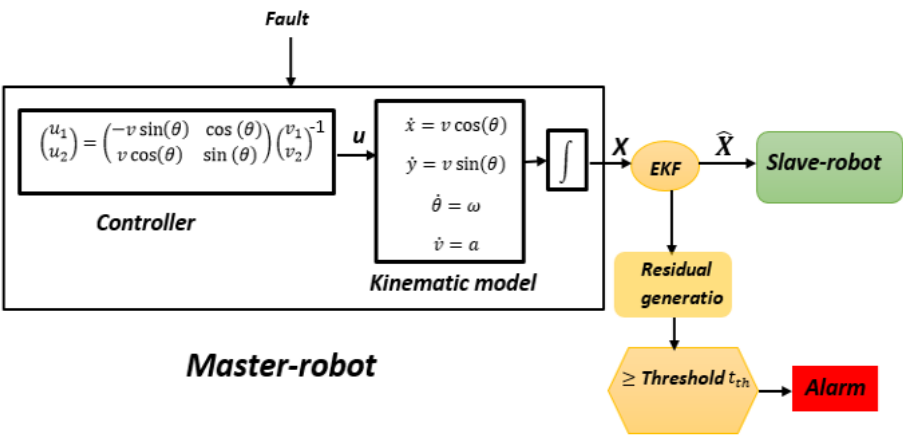


Figure 17. Multi-robot system closed loop

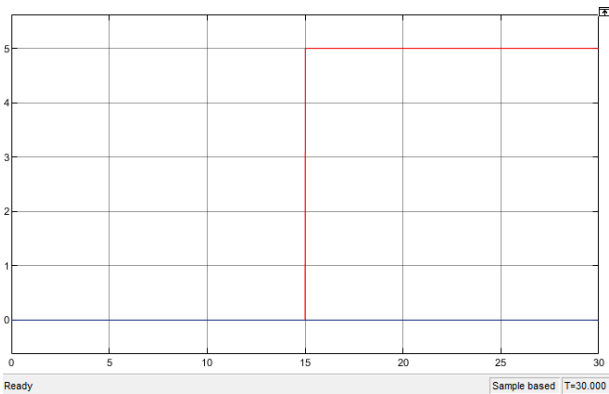


Figure 18. The fault applied in time t=15s

The responses of both the master and the slaves are depicted in the subsequent figures (Figures 19–Figure 21). It is observed that the trajectory of the robots experiences a slight disturbance at a specific time, yet it promptly returns to its intended path without deviating from the desired trajectory.

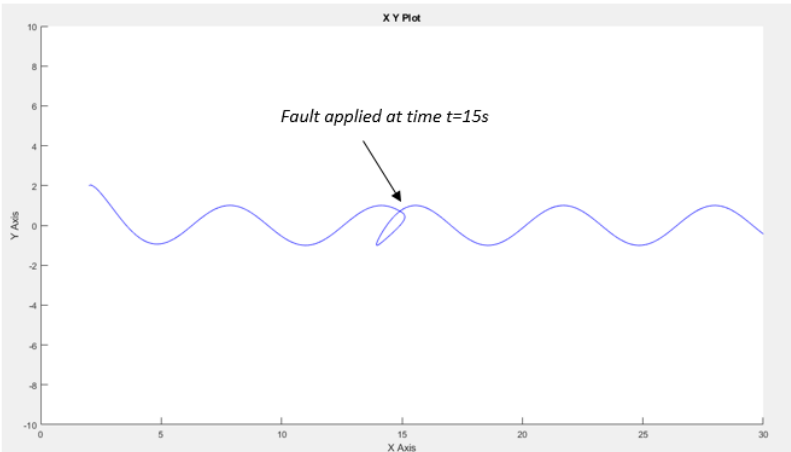


Figure 19. The trajectory of robot-master with fault applied in time t=15s

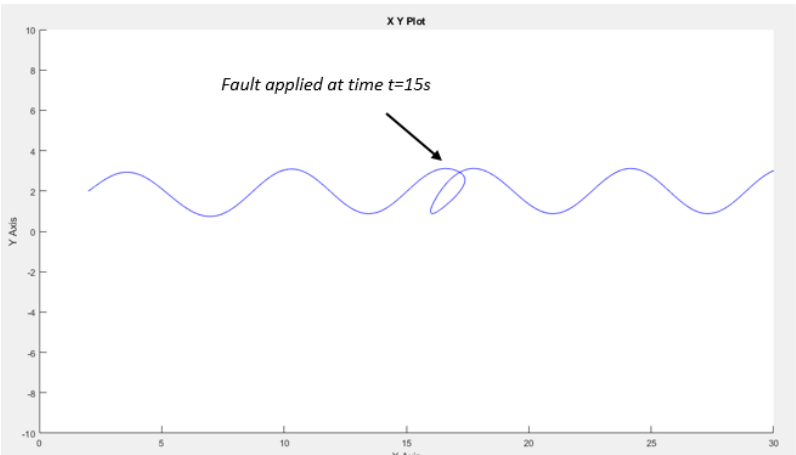


Figure 20. The trajectory of first robot-slave with fault applied in time $t=15s$

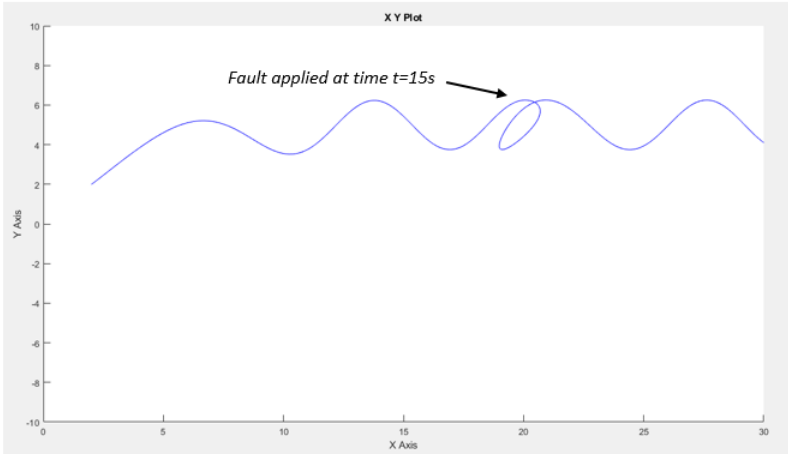


Figure 21. The trajectory of second robot-slave with fault applied in time $t=15s$

The Figures 22 and 23 show the x and y measurement of the desired trajectory, the master and the slaves respectively.

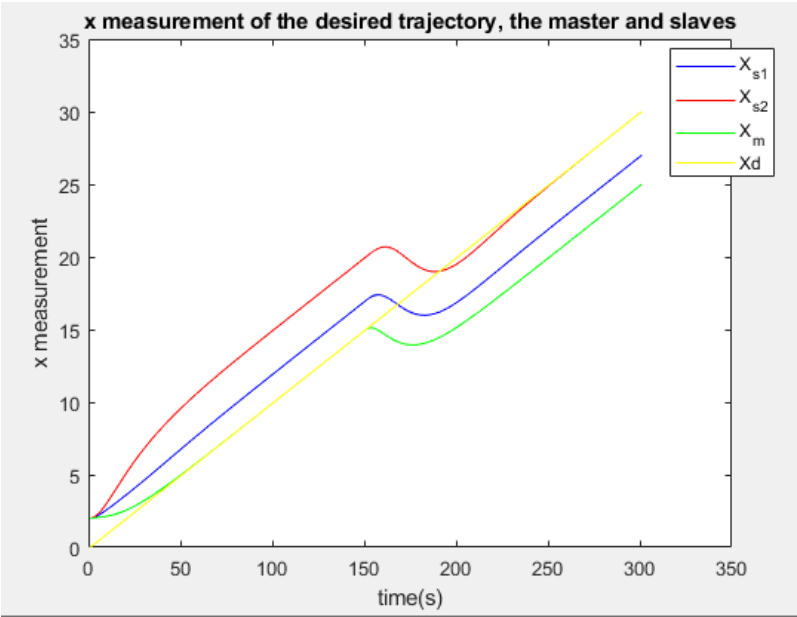


Figure 22. The coordinate x of the desired trajectory, the master and slaves

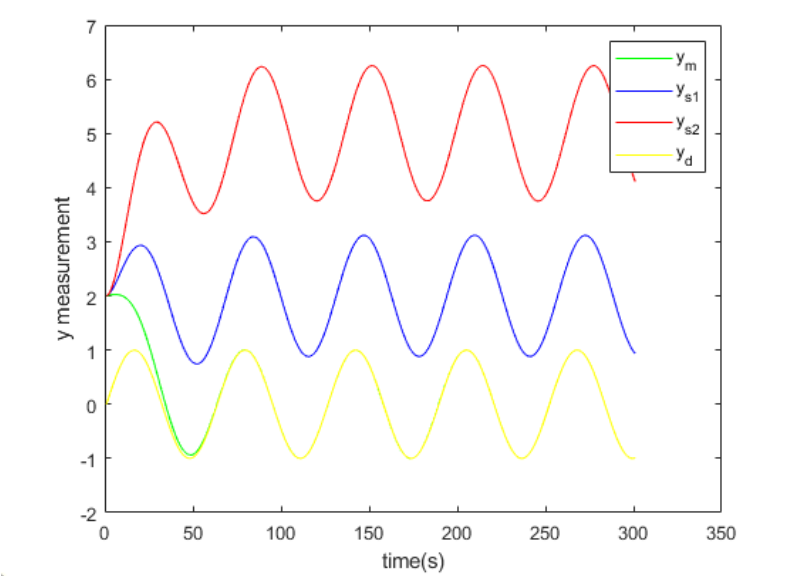


Figure 23. The coordinate y of the desired trajectory, the master and slaves

Furthermore, the simulations of the controllers for both the master and the slaves are illustrated in Figures 24–26 respectively. It is observed that the robot-master adheres to the desired trajectory until time $t = 15s$, at which point it deviates slightly. However, after a few seconds, it resumes following the trajectory initially set.

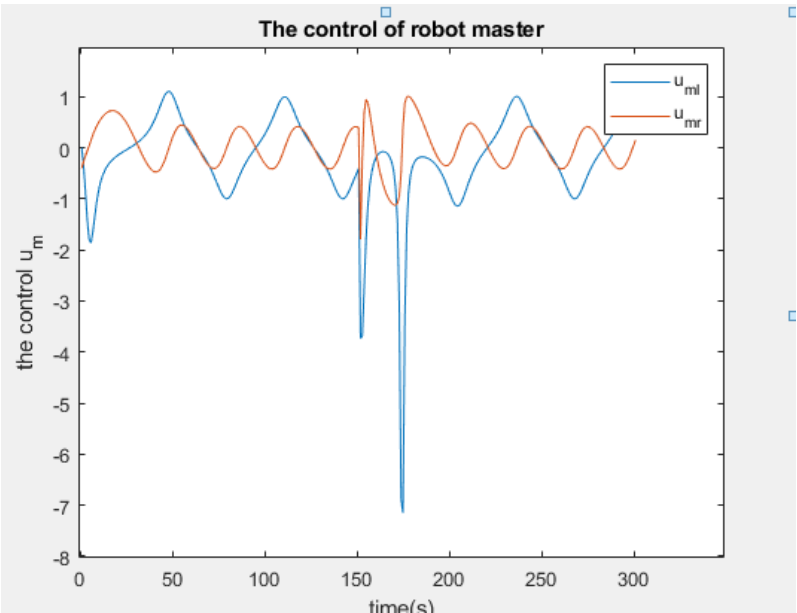


Figure 24. The control of the robot-master

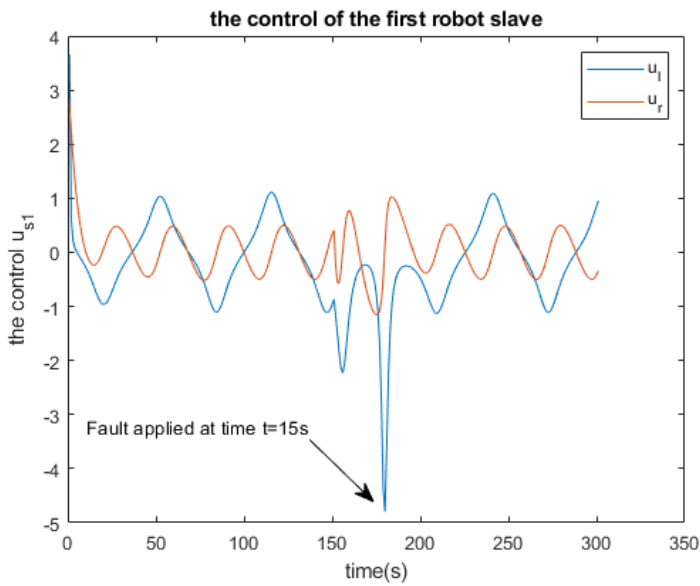


Figure 25. The control of the first robot-slave

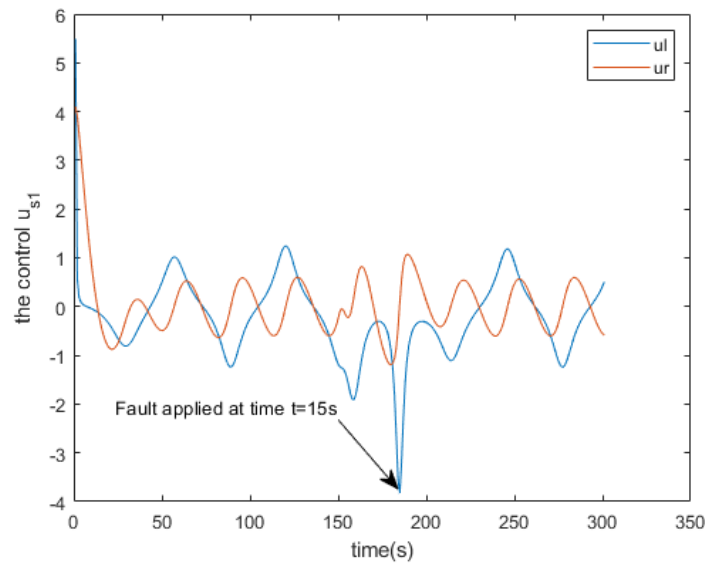


Figure 26. The control of the second robot-slave

Meanwhile, our attention is directed towards the behavior of the robot-slaves, which entails two cases for investigation: firstly, when the robot-slave operates without fault, and secondly, when the robot-slave is subjected to a fault injection. Simulations depicted in Figures 20 and 21 reveal the impact on the functionality of the slave robot. Notably, a minor peak is observed in the slave responses (refer to Figures 22 and 23), following which it swiftly resumes its trajectory.

3.1. Fault Diagnosis

To utilize the Extended Kalman Filter (EKF) for fault detection in a wheeled mobile robot, the following procedural steps are advised:

1- System Modeling: Develop a comprehensive model that characterizes the robot's behavior under both normal operating conditions and various fault scenarios.

2- State Estimation: Implement the EKF to estimate the current state of the robot utilizing sensor measurements.

3- Fault Models: Define specific fault models detailing how each fault impacts the robot's state and sensor measurements.

4- Residual Calculation: Calculate residuals by computing the disparity between actual measurements and estimated measurements, based on the fault models delineated in the preceding step.

5- Fault Detection: Employ statistical tests or threshold-based techniques on the residuals to identify the occurrence of faults.

6- Fault Diagnosis: Initiate the fault diagnosis phase subsequent to the detection of faults.

For establishing thresholds, the system should first undergo simulation under fault-free conditions, followed by simulations with faults. The fault-free simulations aid in determining thresholds through the application of the three-sigma method.

$$\begin{cases} -r = -3\sigma \\ +r = +3\sigma \end{cases}$$

where σ denotes the standard deviation of residual r . As mentioned in the previous section, the fault is said detected when the residual exceeds the threshold more than $|t_{th}|$.

Based on EKF, so in our case we are going to calculate the difference between the values of the robot's state vector and the values of the estimated state vector X_e . So four residuals are generated for each robot.

$$\begin{cases} r_x = x - x_e \\ r_y = y - y_e \\ r_\theta = \theta - \theta_e \\ r_v = v - v_e \end{cases}$$

where the index e means the estimated value. Each residual is compared to the threshold.

Table 1 summarises the signatures of the residuals under fault applied in the robot-master and the slaves robots respectively. The different residuals signatures allow the fault isolation. We notice that when the fault appears in the robot master the other two robots (slaves) are affected. In other side, if we insert the fault in the robot slaves the master is not affected so we conclude that fault affects only the following robot so we can say that the fault propagates in one direction only. As shown in figures (Figures 27–29) fault appears at time $t = 15s$ and is detected and the detection delay does not effect on the system performance.

Table 1. Residual signatures under fault

Fault on	residue of master robot	residue of first slave robot	residue of second slave robot
Master robot	1	1	1
First slave robot	0	1	1
Second slave robot	0	0	1

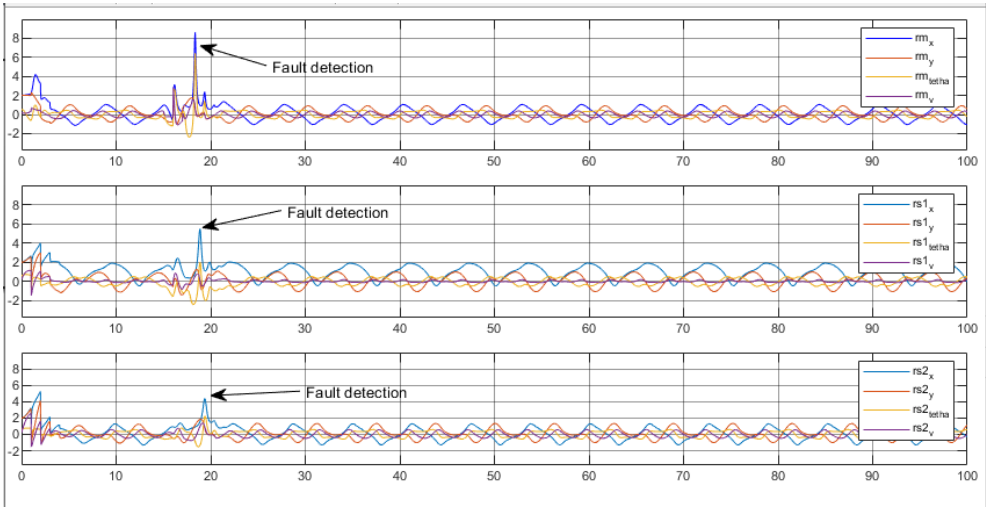


Figure 27. The Residual of the three robots when fault is applied on the master

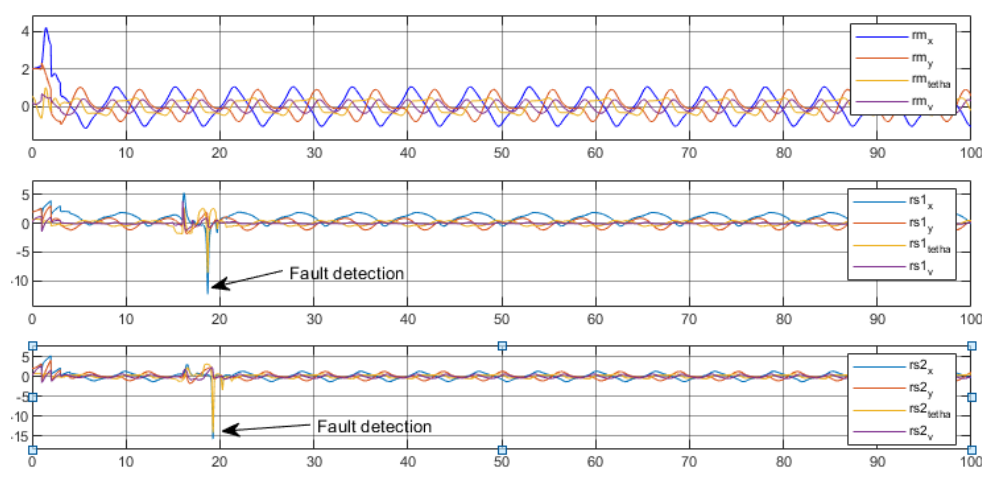


Figure 28. The Residual of the three robots when fault is applied on the first slave

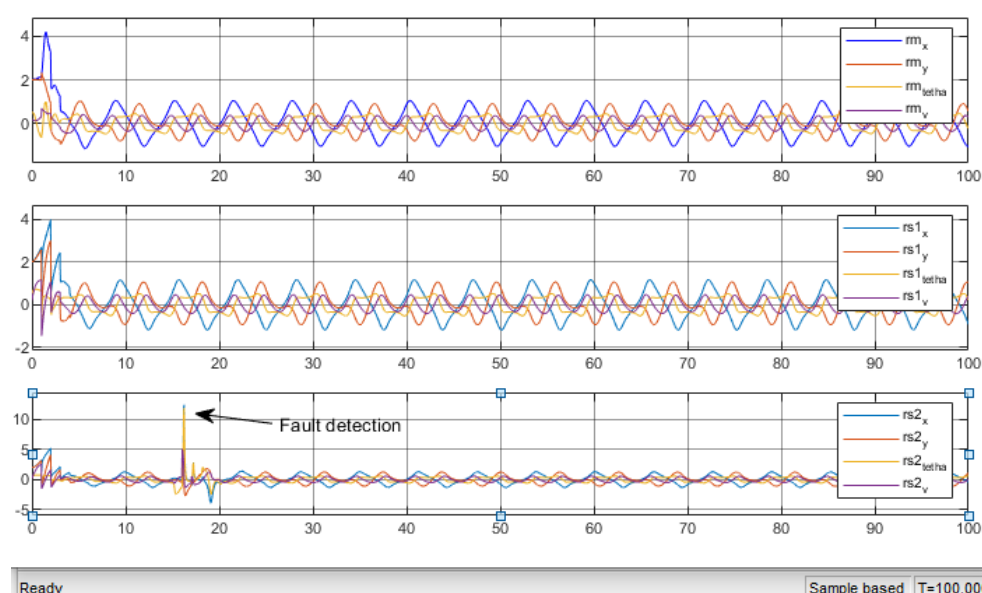


Figure 29. The Residual of the three robots when fault is applied on the second slave

4. Conclusion

The multi-robots system, such as master-slave robots, helps to perform several tasks at the same time which improves the performance of the process. But in case of failure, the diagnostic steps are very important to predict and then solve the problem. So that an extended kalman filter is adopted in order to produce the signatures of the residuals between the multi-robot system and the EKF to detect fault which appear during functioning of the system. The previous simulations show the effectiveness of the proposed methods and how the EKF has managed to achieve the objectives that we set from the beginning.

References

1. Ribeiro, A.F.A.; Lopes, A.C.C.; Ribeiro, T.A.; Pereira, N.S.S.M.; Lopes, G.T.; Ribeiro, A.F.M. Probability-Based Strategy for a Football Multi-Agent Autonomous Robot System. *Robotics* **2024**, *13*. doi:10.3390/robotics13010005.
2. Guo, X.G.; Xu, W.D.; Wang, J.L.; Park, J.H. Distributed neuroadaptive fault-tolerant sliding-mode control for 2-D plane vehicular platoon systems with spacing constraints and unknown direction faults. *Automatica* **2021**, *129*, 109675. <https://doi.org/10.1016/j.automatica.2021.109675>.

3. Ruch, C.; Gachter, J.; Hakenberg, J.; Frazzoli, E. The +1 Method: Model-Free Adaptive Repositioning Policies for Robotic Multi-Agent Systems. *IEEE Transactions on Network Science and Engineering* **2020**, p. 1. doi:10.1109/TNSE.2020.3017526.
4. Engelmann, D.C.; Ferrando, A.; Panisson, A.R.; Ancona, D.; Bordini, R.H.; Mascardi, V. RV4JaCamdash; Towards Runtime Verification of Multi-Agent Systems and Robotic Applications. *Robotics* **2023**, *12*. <https://doi.org/10.3390/robotics12020049>.
5. Jouili, A.; Boussaid, B.; Zouinkhi, A.; Abdelkrim, M. Finite time consensus for multi-tricycle systems under graph theory. 2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2020, pp. 24–29. doi:10.1109/STA50679.2020.9329335.
6. Sayed-Mouchaweh, M.; Billaudel, P. Abrupt and Drift-Like Fault Diagnosis of Concurrent Discrete Event Systems. 2012 11th International Conference on Machine Learning and Applications, 2012, Vol. 2, pp. 434–439. doi:10.1109/ICMLA.2012.157.
7. Khan, A.Q. Observer-based fault detection in nonlinear systems. PhD thesis, Duisburg, Essen, Univ., Diss., 2010, 2010.
8. Sallem, F. Détection et isolation de défauts actionneurs basées sur un modèle de l'organe de commande. PhD thesis, Université Paul Sabatier - Toulouse III, 2013.
9. Mouzakitis, A. Classification of fault diagnosis methods for control systems. *Measurement and Control* **2013**, *46*, 303–308.
10. Fourlas, G.K.; Karras, G.C.; Kyriakopoulos, K.J. Sensors fault diagnosis in autonomous mobile robots using observer — Based technique. 2015 International Conference on Control, Automation and Robotics, 2015, pp. 49–54. doi:10.1109/ICCAR.2015.7166000.
11. Yutian, L.; Jungan, C. Integrated Fault Diagnosis Method of Mobile Robot. Theoretical and Mathematical Foundations of Computer Science; Springer Berlin Heidelberg: Berlin, Heidelberg, 2011; pp. 372–379.
12. Rhouma, T.; Keller, J.Y.; Abdelkrim, M.N. A Kalman filter with intermittent observations and reconstruction of data losses. *International Journal of Applied Mathematics and Computer Science* **2022**, *32*.
13. Ducard, G. Smac-Fdi: A Single Model Active Fault Detection and Isolation System for Unmanned Aircraft. *International Journal of Applied Mathematics and Computer Science* **2015**, *25*. doi:10.1515/amcs-2015-0014.
14. Jouili, A.; Boussaid, B.; Zouinkhi, A.; Abdelkrim, M. 2022 IEEE 21st international Cnference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2022, pp. 237–242.
15. Kia, S.S.; Van Scoy, B.; Cortes, J.; Freeman, R.A.; Lynch, K.M.; Martinez, S. Tutorial on Dynamic Average Consensus: The Problem, Its Applications, and the Algorithms. *IEEE Control Systems Magazine* **2019**, *39*, 40–72. doi:10.1109/MCS.2019.2900783.
16. Boizot, N.; Busvelle, E.; Jean-Paul, G. An adaptive high-gain observer for nonlinear systems. *Automatica* **2010**, *46*, 1483–1488. doi:10.1016/j.automatica.2010.06.004.
17. Bokor, J.; Szabo, Z. Fault detection and isolation in nonlinear systems. *Annual Reviews in Control* **2009**, *33*, 113–123. doi:10.1016/j.arcontrol.2009.09.001.
18. Samia, M.; Graton, G.; El Mostafa, E.; OULADSINE, M.; PLANCHAIS, A. On fault detection and isolation applied on unicycle mobile robot sensors and actuators. 2018 7th International conference on systems and control (ICSC), 2018, pp. 148–153.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.