A Novel Inertia Moment Estimation Algorithm Collaborated with Active Force Control Scheme for Wheeled Mobile Robot Control in Constrained Environments

Regular Paper

Mohammed A. H. Ali[1]*, Muhammad S.A. Radzak[2], Musa Mailah [3], Nukman Yusoff [4], Bushroa Abd Razak [5], Mohd Sayuti Ab. Karim [6], Wadea Ameen [7], Waheb A. Jabbar [8], AbdulRahman A. Alsewari [9], Taha H. Rassem[10] and Abdullah B. Nasser[11]

[1] Department of Mechanical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia, Email address: hashem@um.edu.com.
[2] Faculty of Manufacturing Engineering Technology, College of Engineering Technology, Universiti Malaysia Pahang, 26600 Pekan, Malaysia, Email address: Muhammadsawal_92@yahoo.com.
[3]School of Mechanical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Malaysia, Email address: musa@fkm.utm.my
[4] Department of Mechanical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia, Email address: nukman@um.edu.my
[5] Department of Mechanical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia, Email address: bushroa@um.edu.my
[6] Department of Mechanical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia, Email address: mdsayuti@um.edu.my
[7] Department of Industrial Engineering, Faculty of Engineering and Architecture University of Alyamamah, 13541 Riyadh, Saudi Arabia, Email address: wqaid@ksu.edu.sa.
[8] Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang, 26600 Pekan , Malaysia, Email address: waheb@ieee.org
[9] Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, 26600 Pekan, Malaysia, Email address: alsewari@ump.edu.my
[10] Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, 26600 Pekan, Malaysia, Email address: tahahussein@ump.edu.my
[11] Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, 26600 Pekan, Malaysia, Email address: abdullahnasser@ump.edu.my

*Corresponding author E-mail: hashem@um.edu.com

Received 26 January 2020; Accepted

Abstract

This paper presents a novel inertia moment estimation algorithm to enable the Active Force Control Scheme for tracking a wheeled mobile robot (WMR) effectively in a specific trajectory within constrained environments such as on roads or in factories. This algorithm, also known as laser simulator logic, has the capability to estimate the inertia moment of the AFC-controller when the robot is moving in a pre-planned path with the presence of noisy measurements. The estimation is accomplished by calculating the membership function based on the experts' views in any form (symmetric or non-symmetric) with lowly or highly overlapped linguistic variables. A new Proportional-Derivative Active Force controller (PD-AFC-LS-QC), employing the use of laser simulator logic and quick compensation loop, has been developed in this paper to robustly reject the noise and disturbances. This controller has three feedback control loops, namely, internal, external and quick compensation loops to compensate effectively the disturbances in the constrained environments. A simulation and experimental studies on WMR path control in two kinds of environments; namely, zigzag and highly curved terrains, were conducted to verify the proposed algorithm and controller which was then compared with other existed control schemes. The results of the simulation and experimental works show the capability of the proposed algorithms and the controller to robustly move the WMR in the constrained environments.

Keywords — Laser Simulator Logic (LSL), Noisy and Constraint Environment, Wheeled Mobile Robot (WMR), Active Force Control (AFC).

1. Introduction

Nowadays, mobile robots are widely used in many applications that represent ~~a~~ hazardous, complex, high accurate or challenging tasks, such as aerospace, under-water, military, medicine, inspection, and mining etc. In almost all cases, the robots are supposed to navigate autonomously in restricted environments; which need a highly robust control system to determine its path within the terrain and avoiding obstacles. In some conditions such as road environments, the robot is required to be robustly controlled in the planned path; otherwise it will crash into cars, people or other pedestrians.

A wheeled mobile robot navigation system in restricted environments can accomplish many tasks that must be performed in real-time such as path planning, localization and control of motion. The path planning is the process of finding a path for the robot from the start to the goal destinations while avoiding obstacles and disturbances. The localization is the robot's skill to identify its position within a restricted environment. The control of motion is the most important task in the WMR navigation system that assures the capability of the robots to track the pre-planned path effectively in the restricted environment (Ali et al., 2012a, 2012b, 2014, 2016).

The control of the mobile robot in the restricted areas such as on the road's and in factory's environments during path execution is still a complicated problem in robotics research, since it needs to maintain the tracking errors at the zero levels along the motion. At the same time, the wheel mobile robot must robustly follow the pre-defined path using a suitable control system; otherwise it can cause the robot to crash with other objects. The motion control of the robot in such an environment becomes quite a challenge and a new controller that can able to reject the disturbances in noisy conditions is needed.

The noise in the robotic applications are presented due to the limitations in the sensors measurements and other input devices which can be neither avoided nor controlled its effects inside the controllers, especially in the constrained environments, where the tracking errors must be remained in the zero-level along the motion. In addition, there are some parameters, when designing of the controller's feedbacks such as inertia moments, can't be directly measured and instead ~~it~~ they need to be estimated using offline based AI algorithms (Mailah, 1998), which ultimately adds high risks on adjusting properly the controllers.

Mailah et al. (1998) have found practically that the useful estimated values of the inertia moments are lied in the range of [0.4M-1.4M], where M is the masses of all elements in the robot system. With such range, it is easier to design fuzzy logic membership function as it depends mainly on the total range of fuzzy sets which are later divided into several equally linguistic variables ranges. On other side, other algorithms such as neural network, genetic, iterative-learning, knowledge-based expert, reinforcement and deep learning algorithms need moreover massive samples of input and output for the training propose, which is usually very hard to be determined as the inertia moment can't be directly measured by sensors. Thus, the fuzzy logic algorithm has been ~~is~~ remarked as the best AI algorithm to estimate the inertia moment (Ali and Mailah, 2019a, 2019b).

Despite the preference to utilize the fuzzy logic algorithm for inertia moment estimation, the noise can't be fully eliminated in the constrained environments, due to the highly overlapping of the linguistic variables ranges. Thus there is a need to develop an AI algorithm that can deal well with noise of overlapping between the linguistic variables of the inertia moments sets in the constrained application. This motivated us to develop a new algorithm, called laser simulator logic, to avoid the drawbacks of fuzzy logic algorithm when dealing with the noise coming from highly overlapping of the linguistic variables during estimating of inertia moment in constrained environments.

2. Related works

The related literature on wheeled mobile robot control can be classified into four types, namely, kinematics-based controller, artificial intelligence-based control system, dynamic-based controllers and active force controllers. The main concern will be given to highlight the stability characteristics of these controllers.

2.1 Kinematic Based Controllers

The kinematic based controllers have been widely used to control the position, velocity and acceleration of WMR. A PD controller has been used to avoid obstacles and track the trajectory of the wheel mobile robot with non-holonomic movement (Yang et al., 2016). It has resulted in big tracking errors in the range of 4.5m – 10m if no controller is used. However, the region expanded to 4.9m-8.5m when PD controller is used. A non-linear controller based on a back-stepping strategy has been used to control a virtual wheel mobile robot (Chen and Baoli, 2015) in the desired formation shapes. It uses a cascade system theorem to study the stability of the control system. The results show the capability of the

control system to reduce the settling time to 5 to 10 seconds to perform the desired formation. A tracking control system based on GPS and Real-Time Kinematic (RTK) calculation measurements has been used to track the wheeled mobile robot in a path with skipping and slipping (Low and Wang, 2008). As a result, this controller shows tracking errors in the region of 0 to -1 and 0 to 4 degrees with a settling time of around 20 second.

Yoon et al. (2014) have proposed a control scheme to control an autonomous wheeled mobile robot taking into consideration the slippage and dynamic effects such as fast speed and Coriolis acceleration. The modelling is performed using Lagrange-D'alembert which estimates the relationship between the states and the constrained forces. The back stepping controller is used to track the robot on a particular path in an efficient way. A wheeled mobile robot control system has employed the use of a back-stepping scheme for tracking the wheeled mobile robot in unknown longitudinal and latitude slippage (Cui et al, 2016). Unscented Kalman Filter has beenjh866 incorporated with a low pass filter to estimate the parameters of the slippage in real-time. The results have shown that the control system is capable of reaching asymptotic stability in a short time (the errors are close to zero in 0.2 seconds).

Huang and Zhai (2015) have proposed a non-holonomic control system for a wheeled mobile robot in the presence of external disturbances. In this control, a kinematic controller is used to correct the positions' errors, and a disturbance observer is used to continuously estimate the disturbances. The Lyapunov stability theory was used to study the stability of the control system which shows that the errors between the actual and desired paths tend to be at a zero level in a short time. A kinematic based controller has been used to control a wheeled mobile robot in a cylindrical environment and pipe (Song et al., 2016). The kinematic parameters are integrated with the biaxial clinometer method to enable the robot to move horizontally in the pipes. This control scheme has resulted in quite significant errors but it remains stable after swinging several times. A control of WMR based on an estimation of the slip ratio and longitudinal velocity has been accomplished using sliding mode control and adaptive unscented Kalman filter (Partovibakhsh and Liu, 2015). It has been observed that there are significant errors for wheel angular velocity between the measured and AUKF estimated methods. However, the slip ratio error of the wheel mobile robot using AUKF is stable, starting from time of 2 seconds until steady state. However, the errors of the WMR slip ration are oscillating between 0 and -0.2^0.

2.2 Artificial Intelligence-Based Control System

The artificial intelligence-based control system has been used to control mobile robots due to its capability to make the right decisions. The D-type iterative learning controller has been used in a simulation study for trajectory tracking of WMR (Lu and Fei, 2015) based on kinematics and dynamics parameters. The combination of the iterative learning controller and the dynamics of the robot allows for the robots to be robustly controlled. The results show that the tracking errors become close to zero after the tenth iterations in 10 seconds. A car-like wheeled mobile robot has been controlled in terrain with high uncertainties and noise using RBF neural network (Raeisi et al., 2015). It integrates the RBF neural network with a linear observer to design a controller that can effectively compensate for the uncertainties errors. Lyapunov method has been used to study the stability of the proposed controller which has converged exponentially to the origin. A fuzzy logic control system is employed for controlling a wheeled mobile robot platform to reach the goal with obstacles avoidance (Chiu et al., 2015), and follows a reference trajectory formed by leader-follower steering (Ghiasvand and Alipour, 2013). Both are modelled using Lagrange's Equation to find the relation between the dynamics of the robot and the external disturbances and forces. The results show that the former control system is able to avoid obstacles with small errors. However, in the latter, the tracking errors between the leader and the follower 1,2 and 3 robots are in the region of (0.1 meters, 0.4 meters, 1 meter), respectively.

A non-holonomics WMR path tracking control system in the presence of unknown disturbance and uncertainty has been performed using a computed torque controller and sliding mode controller (Koubaa et al., 2014a). An adaptive-neural control is used for the estimation of disturbances and uncertainties compensation. The Lyapunov stability method (Koubaa et al., 2014b) is used to analyze the response of the control system which resulted in a small error in the range of 0.2 m. A neural network-based adaptive control system has been used for trajectory tracking control (Li et al., 2015) as well as outdoor mobile robot motion control (Guan et al., 2014). In the former, the neural networks control is able to reject the disturbances and approximate the nonlinearity in the system using a radial basis function. However, in the latter, the adaptive controller is able to change the diameter of the WMR wheel based on the road conditions. The results show that the tracking errors of velocities and posture using adaptive neural network controller is converged to zero tracking errors even-though there are external disturbances. A multi-agent robots with a leader-follower consensus problem has been solved using an adaptive control strategy (Chen et al., 2014), which is able to estimate the uncertainties of mechanical parts and control the robot dynamics parameters. During the implementation of this method, there is an overshoot trajectory tracking error which occurs after 2 seconds from settling time, which is compensated quickly by the controller, thereby making it zero. A trajectory tracking control system for controlling a WMR manipulator in the presence of disturbances has been accomplished using an adaptive control system and kinematic controller (Boukattaya et al., 2011). The kinematic controller is used for end-effector position control. However, the adaptive controller is used for uncertainties estimation and disturbances rejection. The results show that the system is stable, and the tracking errors are too small, i.e. near zero (0.001m).

2.3 Dynamic Based Controllers

The Computed Torque Control and nonlinear model predictive control (NMPC) controllers have been used to solve the dynamic model of an omni-wheeled robot (Abhishek and Saha, 2016). In this case, a system identification has been used to find the model parameters with linear least squares technique. The results show that the proposed control system, when the system identification uses 16 values, is able to track a longitudinal and rotational paths with small tracking errors in the region of 5-10mm in the first two seconds which reduces gradually to zero in the steady state. A receding horizon following controller has been proposed to control WMR to eliminate the effect of disturbances and reach asymptotic stability in the following based problem (Liu et al., 2015). It uses two appropriate terminal-penalty and terminal-region to

guarantee the optimum control trajectory with small tracking errors in the region of (0 -0.5m). A sliding mode control scheme has been proposed for WMR motion control in uncertain environments and non-linear trajectories (Zidani et al., 2014). The non-linear control is designed based on the Lyapunov stability method, which resulted in small tracking errors in the region of (-0.5 - 0.2) m. A robust WMR control scheme for rejection of the external disturbances has been introduced using disturbance observer (Huang and Zhai, 2015) and Sliding mode dynamic controller (SMDC) (Koubaa et al., 2014a), which allow for the tracking errors to be compensated and reach zero error in a short time. A sliding mode dynamics has been used for real-time trajectory tracking control for a two-wheeled mobile robot (Mu et al., 2015), which tends to reach zero tracking errors quickly. It has been found that there are small errors between the actual trajectory and reference trajectory and the control system, which becomes stable after 5 seconds.

2.4 Active Force Controller

The AFC generally has two loops; the first loop is used for kinematic parameters control and another loop for dynamics control of a mobile robot. The active force control has been used to reject the unwanted disturbance of frictional force in the pipe (Sabzehmeidani et al., 2011), track the wheeled mobile robot in a pre-planning path in a difficult environment (Ali et al., 2015), remove the effect of disturbance in the system in the dynamic system (Mailah et al., 2015), and reject the noise (Abdullah et al., 2015).

The Proportional-integral-derivative (PID), Active Force Control and sliding mode controller (SMC) (Sabzehmeidani et al., 2011) have resulted by the tracking errors that are oscillating clearly in 1 to 10 seconds and then become stable at zero levels. Resolved Acceleration Control (RAC) with Active force control (AFC) (Ali et al., 2015, 2019a) have figured out

Table 1 A comparison between related works' controllers

Author	Controller	Stability Characteristics
Yang *et al.,* 2016;	PD controller	big tracking errors in the range of (4.5m – 10m)
Chen *et.al.* , 2015	Non-linear controller based on back-stepping	5 - 10 seconds to reach the desired position
Low & Wang, 2008	GPS and Real-Time Kinematic	Errors in the region of 0 to -1 and 0 to 4 degree with a settling time 20 seconds
Cui *et.al*, 2016	Back stepping controller with Unscented Kalman Filter	Reach asymptotic stability in a short time; where errors become close to zero in 0.2 seconds
Huang *et.al.* 2015	Non-holonomic control, kinematic controller	Errors between actual and desired paths tend to be in a zero level in a short time
Song et al., 2016	kinematic based controller	Big errors but it still remain stable after swinging several times
Partovibakhsh *et.al.* , 2015	Sliding mode control and adaptive unscented Kalman filter	Errors of mobile robot slip ration is oscillating between 0 and -0.2^0.
Lu *et al.*,, 2015	D-type iterative learning controller	Close to zero after tenth iterations within 10 seconds
Raeisi, *et al.,* 2015	RBF neural network	Errors are close to zero in 0.5-0.8 seconds
Chiu *et al.,* 2015	Fuzzy logic control	Mobile robot is able to avoid obstacles with small tracking errors
Ghiasvand *et al.*,, 2013	Fuzzy logic control and using Lagrange's equation	Tracking errors between the leader and follower robots are small (0.1-0.4m)
Koubaa *et al.,* 2014b	An adaptive-neural control	Small errors in the range of 0- 0.2 m
Li *et al.,* 2015 and Guan et al., 2014	Neural network-based adaptive control, Adaptive control strategy	Tracking errors of velocity and posture using adaptive neural network controller is converge to zero tracking errors
Chen *et.al.,* 2014	Adaptive control strategy	An overshoot of trajectory tracking errors occurs after settling time by 2s
Boukattaya *et.al.,* 2011	Adaptive control and kinematic controller	Stable and the tracking errors are too small, i.e. near to zero (0.001m)
Liu *et.al.,* 2015	Horizon following controller	Small tracking errors in the region of (0 -0.5m)
Huang et al., 2015 and Koubaa *et al.*,, 2014a	disturbance observer, Sliding mode dynamic controller	Compensate the tracking errors and reach zero error in a short time
Mu *et.al.,* 2015	Sliding mode dynamics	Tends to reach zero tracking errors in a short time
Zidani,*et.al*, 2014	Sliding mode control and non-linear control	Small tracking errors in the region of (-0.5 - 0.2) m
Ali *et al.,* 2015 and 2019a	Resolved Acceleration Control (RAC) with Active force control (AFC)	No difference between reference and actual paths when there is no disturbance applied on RAC-AFC. Small errors in power 10^-2mm is occurred when applying disturbances.
Musa *et.al,* 2012	Feedforward active force control (AFC) with Robust controller	Tracking error of AFC oscillates between -0.01 to 0.02 rad
Sabzehmeidani *et al.,* 2011	PID, Active Force Control and sliding mode controller (SMC)	Tracking errors are oscillating clearly in a time 1 to 10 second and then become stable at zero level
Abdullah *et.al.,* 2015	Active force control (AFC) with computed torque control	Tracking errors in AFC-computed torque controller oscillates between -0.04 to 0.03 rad

that there is no difference between the reference and actual paths when there is no disturbance for the RAC-AFC. However, the tracking error has a slight difference between reference and actual paths with small errors in the power of 10^-2mm when a constant disturbance is used.
A feedforward active force control (AFC) with a robust controller (Mailah et al., 2012) and computed torque control (Abdullah et al., 2015) show that the tracking errors of AFC oscillates between -0.01 and 0.02 rad.
However, the tracking errors in the AFC-computed torque controller oscillates between -0.04 and 0.03 rad, thereby AFC is performing better than AFC-computed torque controller.
The comparison between the related works in wheel mobile robot control is illustrated in Table 1:

3. Proposed Controller

This main contribution of this work is to develop a novel AI algorithm for estimating the inertia moment in a noisy environment. In contrast with the current noise-elimination algorithms, it is able to deal with the unexpected noise that are occurred during real-time exploration and thus reduce the calculated errors. With such an advantage, a new AFC controller equipped with the proposed AI based inertia moment estimation is able to control WMR robustly during movement in the constrained environment. In fact, the current active force control is working better than other controllers, especially in the presence of disturbances (Ali et al., 2015; Mailah et al., 2012; Abdullah et al., 2015; Sabzehmeidani et al., 2011). However it still has some noises especially with real-time applications, where the system must compensate the disturbances well.
The sources of noises in Active Force Control (AFC) come from the sensors' measurements such as encoders, accelerometer and gyroscope which are almost fused together and increase the noises. The other source of uncertainties is the inertia moment estimations that are calculated using artificial intelligence and can't be measured. In fact, inertia moment estimations are subjected to the AI algorithm performance. Thus, it is required to enhance the current AFC to reject the noises and estimate the inertia matrix that will be used to control the torques affected on the wheels well. In order to solve the latter's problem, a new algorithm, called Laser Simulator Logic, for estimation of WMR inertia moment in a noisy environment has been introduced in this paper.
Several methods have been used for estimating the inertia moment of the AFC controller. The conventional methods that have been widely used to estimate the inertia moments include crude-approaximation (Hewit and Burdess, 1981), iterative-learning (Mailah, 1998), neural-network (Kwek et al., 2003), knowledge-based expert (Pitowarno et al., 2003) and fuzzy-logic (FL) (Ali and Mailah, 2019a, 2019b). Most of these methods calculate the inertia moment offline and thus they are not suitable in the noisy system. In addition, all the previous methods except fuzzy logic need prior knowledge about some samples of input and output of the system, which are almost not available and making it difficult to use these algorithms in noisy environments. Although fuzzy logic seems to be the best choice in comparison with other algorithms, it is suffering from the following issues:
1. The linguistic variable ranges in most of the current fuzzy logic systems are chosen arbitrarily by users, in which the inference of each variable often starts from the middle of each other~~'s~~ as shown in Fig. 4. The fuzzy logic becomes so noisy when there is an overlapping/ inference in the linguistic variables starting before the middle of each linguistic variables as shown in Figs. 2 and 8 (a), where the accumulative membership values across all linguistic variable for a specific crisp input becomes higher than 1. It seems that there are some regions in the linguistic variables where the user is making sure that they are not affected by other linguistic variables. This conflicts with the basic idea of fuzzy logic that considers every-thing as fuzzy and get different membership value.
2. Fuzzy logic type-1 fails to deal with noises at all. However, type-2 can deal with noises associated with each linguistic variable by taking into consideration a range of values suggested by the experts. However, none of them is dealing with noises coming from the inference between the linguistic variables. This occurs when there is a wide-difference in the decisions taken from experts. In others words, the experts in the fuzzy type-2 have mapped the linguistic variable into multi-values of membership function to avoid the noise and uncertainties in each linguistic variable. However, they didn't take into consideration the effect of the noise that are coming from the high degree of inference between the linguistic variables.
3. The capability of learning in the fuzzy logic algorithm is impossible. Hence, fuzzy logic in its current form has to be integrated with other algorithms to enable learning such as the neural networks (Lotfi and Tsoi, 1996), genetic and reinforcement algorithms (Chiang et al., 1979) and deep learning (Godfrey and Gashler, 2017) which are complex and compute-intensive. Thus, a simple method based on solving a set of equations will be introduced to enable learning in the laser simulator logic.
To solve the 1st and 2nd problems, we aim to create a new algorithm, called laser simulator logic, that uses a dynamic-scale for membership values *i.e.* normalization. This means that membership value is still located in the range $(x) \in [0\ 1]$, but with a different scale of membership value for each crisp value based on the contribution of linguistic variables at the specific regions, as shown in Fig. 2.
The methodology activities for performing this work are shown in Fig. 1.

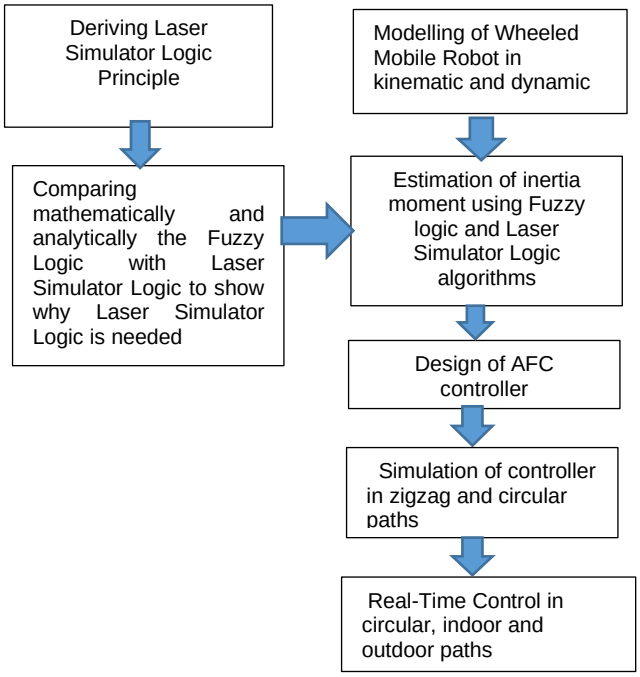


Fig. 1: The methodology activities of presented work in this paper

In this methodology, the laser simulator logic has been derived first in Section 4 and the proven through comparison with the fuzzy logic algorithm in Section 5. The kinematic and dynamic equations of the four-wheeled (two differential wheels and two castor wheels) have been derived in Section 6. An active force controller has been developed in which the estimation of the inertia moment is calculated using a proposed laser simulator logic in section 6. The simulation and real-time experimental works in Sections 6 and 7, have respectively been conducted to prove that the controller with the proposed algorithm is performing better than other controllers.
In general, the contribution of this paper can be briefly stated as follows:

- To develop a new AI algorithm for the estimation of the WMR inertia moment in the noisy environment which is used in the Active Force Control to estimate the applied torque by multiplying the inertia moment with the measured acceleration.
- To create a new control scheme for noise rejection of the current Active Force Control (AFC) through adding a quick compensation loop to compensate for the acceleration at the beginning of the movement in the current AFC.

4 Laser Simulator Logic Principle

This paper is mainly aimed at developing an efficient technique, called Laser Simulator Inference System, by calculating the inertia moment in a noisy environment and dealing with high interference between linguistic variables. This algorithm is different than Laser simulator for path planning (Ali et al., 2019b). The laser simulator for path planning (Ali et al., 2019b) is a novel algorithm that has been used to determine the robot's path within the constrained environment, which has been successfully implemented in the road roundabout area as the main area of constrains in the roads. However, the Laser Simulator Logic algorithm presented in this paper is a novel inference system based on an artificial intelligence algorithm that can deal well with the noise coming from a high degree of overlapping between the linguistic variables. In other words, it helps to avoid the drawbacks of fuzzy logic when there is a significant overlapping between linguistic variables that will be discussed later in detail. The difference between laser simulator and laser simulator logic is illustrated in Table 2.

As shown in Fig. 2, such a high inference system will cause-the accumulative membership values for a crisp input/output to exceed 1 as shown in cases x_1, x_2, x_3 and x_4 in Fig. 2.
The membership value of the crisp input x_1 across all three linguistic variables has an accumulative membership value higher than 1. Thus, there is a need to make a dynamic range for values of 1 when there is a high inference between linguistic variables in membership functions as shown in Fig. 2, to accommodate the noises coming from the high inference/overlapping of linguistic variables, which can vary from a low level in region 1 until the high level in Region 3. This can be accomplished by dividing the universe of discourse into regions that have a different scale of membership based on the inference between the linguistic variables e.g. there are 5 regions in Fig. 2.

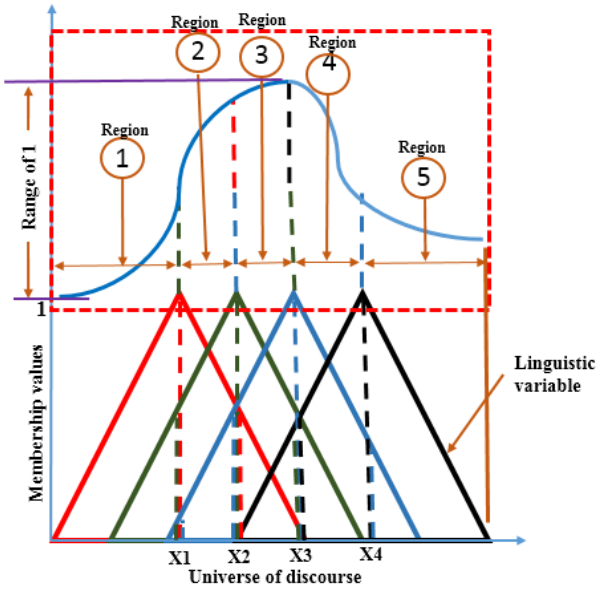


Fig. 2 High inference membership function

Most of the users utilize the fuzzy logic system in the form of identical triangular, Gaussian or trapezoidal membership functions as shown in Fig. 4, in which the linguistic variable can be generally divided into two parts,
The first one is located before the middle of the linguistic variable, and the other one is located after the middle; both give membership values that are proportional to the input location.

Table 2: Shows the difference between laser simulator based path planning and laser simulator logic

Features	**Laser Simulator for Path Planning**	**Laser Simulator Logic**
Algorithm classification	Path planning algorithm	Inference system algorithm
Purpose	Find the most useful path planning of robot	Deal with noise coming from highly overlapped linguistic variables in the inference system.
Principle of work	It generates series of points as horizontal and vertical to determine the best path	It calculates the membership function with any degree of overlapping and guarantees to make the accumulative membership function values for the crisp input always less or equal than 1
Applications	Path planning in Robotic	Control, classification and decision making of any noisy systems
Degree of intelligence	No intelligence since it determines the path according to the generated points between the road curbs *e.g* middle	It uses a certain degree of intelligence to come up with the right decision when dealing with the noise of the system
Can each one replace the other	No	No

In fact, the laser simulator logic depends mainly on these proportional properties between the membership values and its location in the range of input/output sets to form the input/output membership functions as shown in Fig. 5. In other words, it will determine the membership values of each linguistic variable based on its position in the range of linguistic variable in the input sets as seen in Eq. 1 and 2; and apply the implication to the output set based on the fuzzy rules of the system.

Let~~'s~~ us consider a fuzzy set R as a set function on the universe of discourse X, with the membership function $\mu_i: X \to [0\ 1]$, which can be written as: $R = \{(x, \mu_i(x)) \mid x \in X\}$

The equations of laser simulator will be derived in 2 cases: - (1) memberships without inference between the linguistic variables, (2) membership with lowly/highly overlapped linguistic variables. Fig. 3 demonstrates the behaviour of LS with overlapped linguistic variable.

Where a_i, b_i and m_i are the starting, end and middle points of the linguistic variable. In other words, in the case of single linguistic variable, two ranges in the universe of discourse ~~are~~ existed with maximum range values equal to m_i-a_i and b_i-m_i. Eq. 1 can be rewritten as Eq. 2

$$\mu_i(x) = \begin{cases} 0 & x < a_i \\ \frac{x-a_i}{(x-a_i)+(m_i-x)} & a_i < x < m_i \\ \frac{b_i-x}{(b_i-x)+(b_i-m_i)} & m_i < x < b_i \\ 0 & x > b_i \end{cases} \quad (2)$$

Thus, in the case that there is no inference/overlapping between the linguistic variables and only one linguistic variable is fuzzified as shown in Fig. 5, Eq. 1 and 2 can be written for x_1 and x_2 (see Fig. 5):

$\mu_{x1} = \frac{x1-a}{(x1-a)+\lvert(m-x1)\rvert}$ and $\mu_{x2} = \frac{b-x2}{(b-x2)+\lvert(m-x2)\rvert}$

where x_1 and x_2 are arbitrary crisp inputs values located in the range of the linguistic variable.

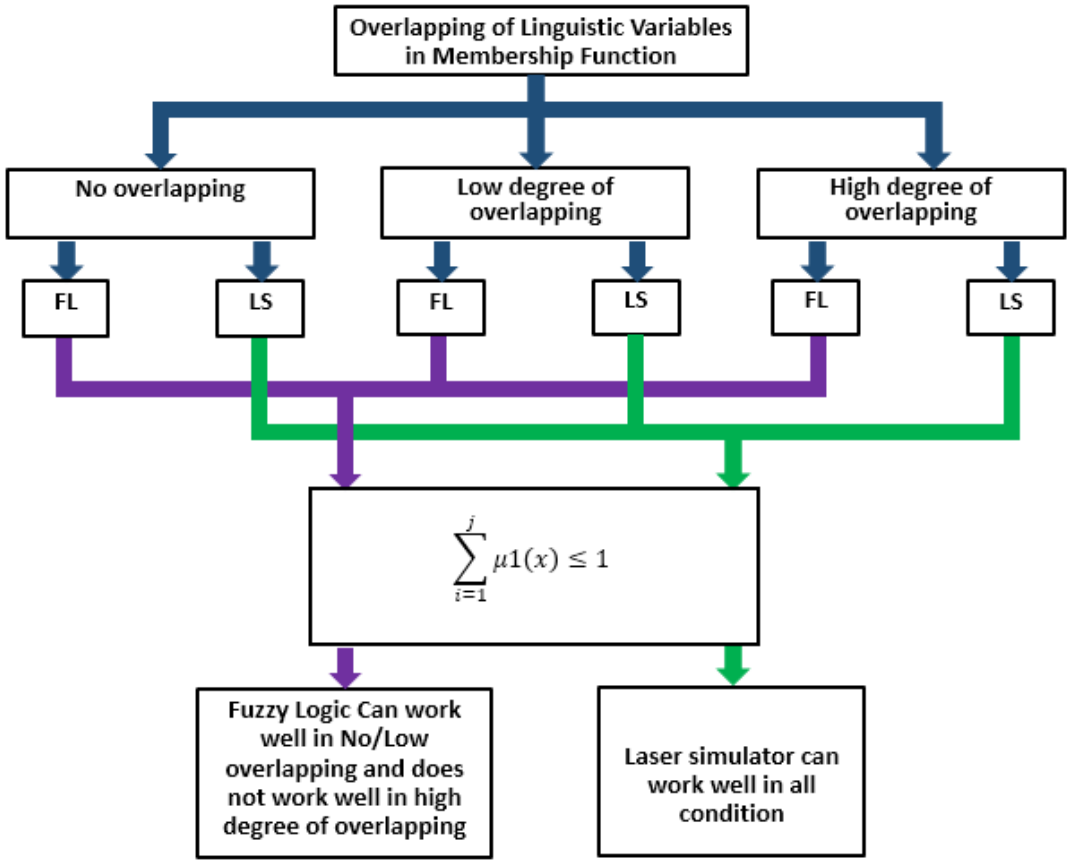


Fig. 3 Fuzzy Logic and Laser Simulator performance with overlapped linguistic variables of membership functions

4.1 Memberships without Inference between Linguistic Variables:

In this case, the membership values of laser simulator can be defined simply as a proportional ratio to its position in the membership function of the linguistic variable range as shown in Fig. 5 and Eqs. 3.1-2:

$$\mu_i(x) = \begin{cases} 0 & x < a_i \\ \frac{x-a_i}{(m_i-a_i)} & a_i < x < m_i \\ \frac{b_i-x}{(b_i-m_i)} & m_i < x < b_i \\ 0 & x > b_i \end{cases} \quad (1)$$

In the implication process of laser simulator logic, the input's membership value calculated by Eq. 1 and 2 will be applied on the linguistic variable of the output fuzzy set as stated in the fuzzy rules with a proportional ratio to its range of this linguistic variable as in Eq. 3.

$$y_i = \mu_i(x) \times Y_i \quad (3)$$

where Y_i is the output linguistic range, y_i is the result of the implication process. The crisp output is then determined by:

$$y = \frac{\sum_{i=1}^{n} y_i}{n} \quad (4)$$

where n is the number of output linguistic variables in the rules' consequents.

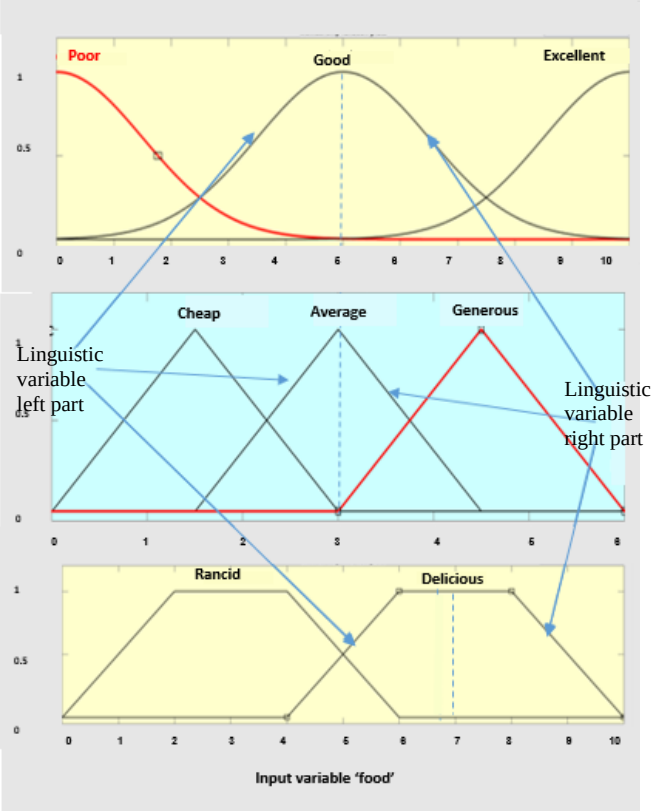


Fig. 4 Common membership functions

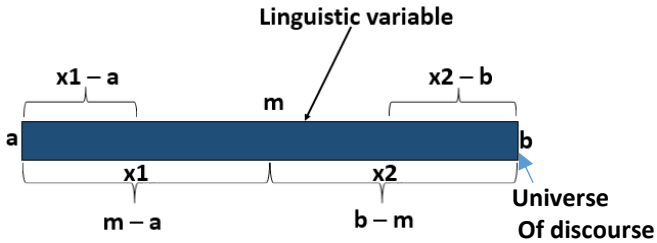


Fig. 5 Basic of laser simulator logic for a single linguistic variable

In the implication process of laser simulator logic, the input's membership value calculated by Eq. 1 and 2 will be applied on the linguistic variable of the output fuzzy set as stated in the fuzzy rules with a proportional ratio to its range of this linguistic variable as in Eq. 3.

$$y_i = \mu_i(x) \times Y_i \quad (3)$$

where Y_i is the output linguistic range, y_i is the result of the implication process. The crisp output is then determined by:

$$y = \frac{\sum_{i=1}^{n} y_i}{n} \quad (4)$$

where n is the number of output linguistic variables in the rules' consequents. Fig. 6 shows an example of the implication of LS inference system which consists of:

- 1 fuzzy input set with 2 linguistic variables
- 1 fuzzy output set with 2 linguistic variables

The inference rules are:

If X is A Then Y is C

If X is B Then Y is D

Based on Eq. 4, the crisp output y is calculated as follows:

$y = \frac{y_1+y_2}{2}$

4.2 Memberships with Inference between Linguistic Variables:

Let~~'s~~ us assume that we have *j* overlapped linguistic variables in a membership function *X*. If $a_1, a_2, a_3, \ldots\ldots, a_j$ are the starting points of the overlapped linguistic variables, $b_1, b_2, b_3, \ldots\ldots$, are the end points of the overlapped linguistic variables and $m_1, m_2, m_3, \ldots\ldots, m_j$ are the middle points of the overlapped linguistic variables, as shown Fig. 7. The accumulative membership function values for the crisp input *(x)* across all overlapped linguistic variables must achieve Eq. 5:

$$\sum_{i=1}^{j} \mu_i(x) \leq 1 \quad | x \in X \quad (5)$$

Two types of inference between linguistic variables are figured out, namely low and high inference systems. The low degree of inference occurs when the linguistic variable starts from the middle or after the middle of the preceding linguistic variable. However, the high degree of inference between the linguistic variables occurs when the current linguistic variable starts before the middle of the preceding one.

In the low inference, both fuzzy logic and laser simulator can guarantee Eq. 5, along with the universe of discourse. However, fuzzy logic types 1 and 2 can't guarantee Eq. 1 for some crisp inputs in a highly overlapped inference system as will be discussed with depth in Section 5. In fact, laser simulator logic can achieve Eq.5 and deal well with highly overlapped inference system, as will be explained in this section.

The equations of laser simulator logic for lowly and highly overlapped membership functions will be derived as follows:

4.2.1 Low degree of inference between linguistic variables:

In this case, the membership value of the crisp input is determined as a proportional ratio to its position in the linguistic variable range without any consideration to other linguistic variables as shown in Fig. 5 using Eq. 1 and 2. So if there is a lowly overlapped area between linguistic variables in the membership functions, the accumulative membership values across all linguistic variables for a crisp input are calculated using Eq. 6:

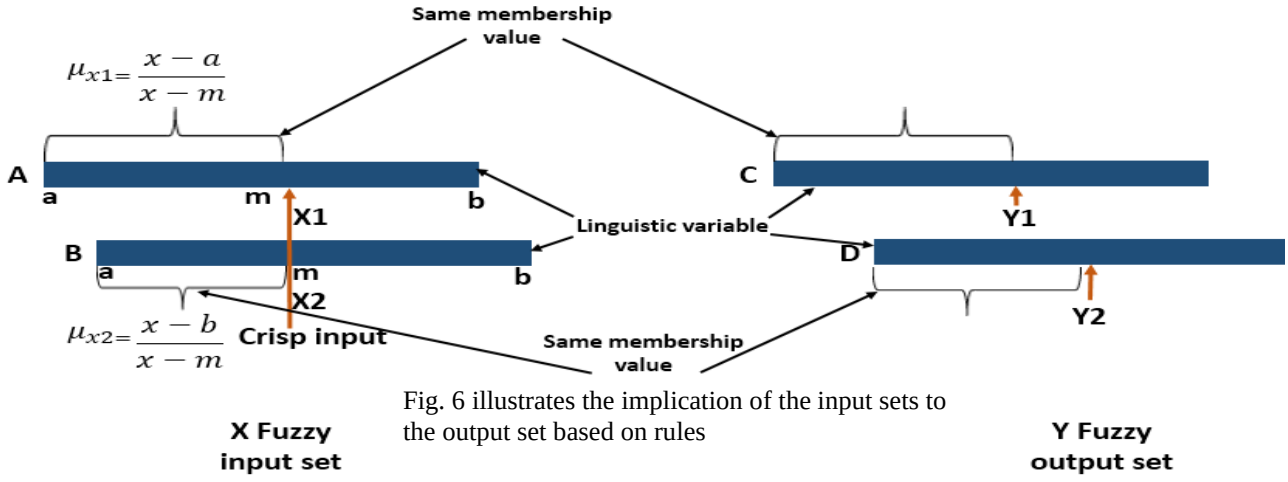


Fig. 6 illustrates the implication of the input sets to the output set based on rules

$$\sum_{i=1}^{j}\mu_i(x)=\begin{cases}0 & x<a_1\\ \frac{x-a_1}{(x-a_1)+(m_1-x)} & a_1<x<m_1\\ \frac{b_1-x}{(b_1-x)+(b_1-m_1)}+\frac{x-a_2}{(x-a_2)+(m_2-x)}+\cdots+\frac{x-a_j}{(x-a_j)+(m_j-x)} & m_1<x<b_1\\ \frac{b_2-x}{(b_2-x)+(b_2-m_2)}+\frac{x-a_3}{(x-a_3)+(m_3-x)}+\cdots+\frac{x-a_j}{(x-a_j)+(m_j-x)} & b_1<x<m_2\\ \vdots & \\ \frac{b_{j-1}-x}{(b_{j-1}-x)+(b_{j-1}-m_{j-1})}+\frac{x-a_j}{(x-a_j)+(m_j-x)} & m_{j-1}<x<b_{j-1}\\ \frac{b_j-x}{(b_j-x)+(x-m_j)} & m_j<x<b_j\\ 0 & x>b_j\end{cases} \quad (6)$$

In fact, the accumulative value of membership is just a summation of individual membership values for the linguistic variables in this membership function, which looks similar to a triangular membership function, where Eq. 5 is achieved along with the universe of discourse.

4.2.2 High degree of inference between linguistic variables:

This is the bottleneck case that presents the novelty of laser simulator logic algorithm. In such highly overlapped membership function, the membership value of the crisp input is determined as a proportional ratio not only to its position in a certain linguistic variable range, but also to all overlapped linguistic variables within a specific range. In fact, the accumulative membership value in fuzzy logic across all linguistic variable for a crisp input becomes sometimes higher than 1, as shown in Figs. 2, 7(a) and 8 (b) and will be mathematically proven in Section 5. This is due to that the membership value in fuzzy logic types 1 and 2 is calculated individually for each linguistic variable and doesn't take into consideration that the summation of membership values for a certain crisp input must be less than 1 as shown in Eq. 5. Thus, the laser simulator membership value of highly overlapped linguistic variables depends on all overlapped linguistic variables in a specific range i.e. the term $|m_i\text{-}x|$ in Eq. 1-4 and 6 have to be replaced by a highly overlapped range ΔS as in Eqs. 7-16, to assure that the accumulative values of membership don't exceed 1 at any case.

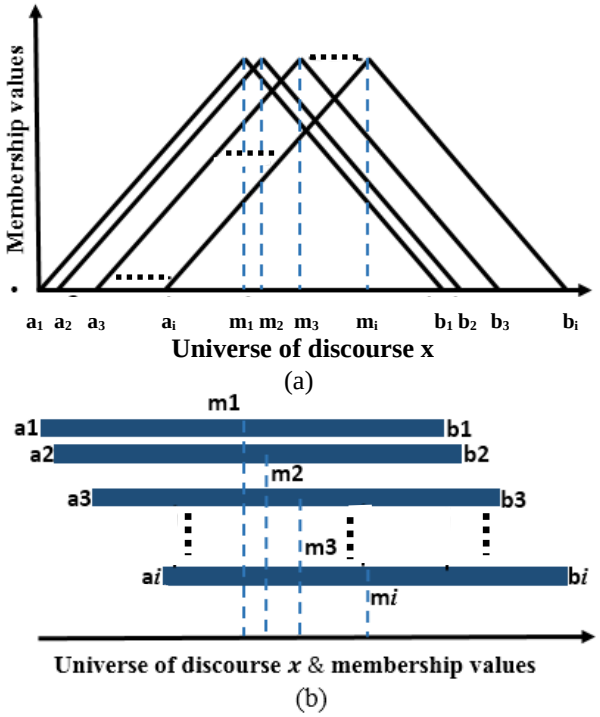


Fig. 7 Highly overlapping range of linguistic variables in membership functions: (a) traditional triangle (b) laser simulator logic

The highly overlapped range ΔS can be defined:

$$\Delta S=\sum_{i=1}^{j}\Delta S_i$$

$$=\begin{cases}0 & x\le a_1\\ (x-a_1)+(x-m_1) & a_1<x\le a_2\\ (x-a_1)+(x-a_2)+\cdots+(x-a_j) & a_2<x\le m_1\\ (b_1-x)+(x-a_2)+\cdots+(x-a_j) & m_1<x\le b_1\\ (b_2-x)+(x-a_3)+\cdots+(x-a_j) & b_1<x\le b_2\\ \vdots & \\ (b_{j-1}-x)+(x-a_j) & b_{j-2}<x\le b_{j-1}\\ (b_j-x)+(x-m_j) & b_{j-1}<x\le b_j\\ 0 & x\ge b_j\end{cases} \quad (7)$$

Note: the overlapping in Eq. 7 had not occurred in the1st two cases ($x\le a_1$ *and* $a_1<x\le a_2$) and last two cases ($b_{j-1}<x\le b_j$ and $x\ge b_j$). In Eq. 7, the universe of discourse has been divided into several ranges located between a_i, b_i and m_i, in which ΔS values assure that the accumulative values of membership in each range don't exceed 1. The accumulative membership value of crisp x in highly overlapped inference system is then calculated using Eq. 8:

$$\sum_{i=1}^{j}\mu_i(x)=\begin{cases}0 & x\le a_1\\ \frac{x-a_1}{(x-a_1)+(m_1-x)} & a_1<x\le a_2\\ \frac{x-a_1}{\Delta S}+\frac{x-a_2}{\Delta S}+\cdots+\frac{x-a_j}{\Delta S} & a_2<x\le m_1\\ \frac{b_1-x}{\Delta S}+\frac{x-a_2}{\Delta S}+\cdots+\frac{x-a_j}{\Delta S} & m_1<x\le b_1\ \&\ x\le m_2\\ \frac{b_1-x}{\Delta S}+\frac{b_2-x}{\Delta S}+\cdots+\frac{x-a_j}{\Delta S} & m_1<x\le b_1\ \&\ x\ge m_2\\ \frac{b_2-x}{\Delta S}+\frac{x-a_3}{\Delta S}+\cdots+\frac{x-a_j}{\Delta S} & b_1<x\le b_2\ \&\ x\le m_3\\ \frac{b_2-x}{\Delta S}+\frac{b_3-x}{\Delta S}+\cdots+\frac{x-a_j}{\Delta S} & b_1<x\le b_2\ \&\ x\ge m_3\\ \vdots & \\ \frac{b_{j-1}-x}{\Delta S}+\frac{x-a_j}{\Delta S} & b_{j-2}<x\le b_{j-1}\ \&\ x\le m_j\\ \frac{b_{j-1}-x}{\Delta S}+\frac{x-a_j}{\Delta S} & b_{j-2}<x\le b_{j-1}\ \&\ x\ge m_j\\ \frac{b_i-x}{(b_i-x)+(x-m_i)} & b_{j-1}<x\le b_j\\ 0 & x\ge b_j\end{cases} \quad (8)$$

Eq. 8 is useful when there are a few linguistic variables with high overlapping. However, it becomes difficult when there are more than four-overlapped linguistic variables. Thus, to decrease the numbers of possible ranges in Eq. 7, ΔS can be briefly written as in Eq. 9 to easily find the membership value for each linguistic variable, no matter whether the crisp value x is located before or after the middle of the linguistic variable m_i.

This can be accomplished using the following expression:

$$\min\big((x-a_i),(b_i-x)\big)=\begin{cases}x-a_i & a_i<x<m_i\\ b_i-x & m_i<x<b_i\end{cases}$$

So Eq. 7 can be rewritten as in Eq. 9 as follows:

$$\Delta S=\sum_{i=1}^{j}\Delta S_i=\begin{cases}\sum_{i=1}^{j}\min\big((x-a_i),(b_i-x)\big) & a_2<x\le b_1\\ \sum_{i=2}^{j}\min\big((x-a_i),(b_i-x)\big) & b_1<x\le b_2\\ \vdots & \\ \sum_{i=j-1}^{j}\min\big((x-a_i),(b_i-x)\big) & b_{j-2}<x\le b_{j-1}\end{cases} \quad (9)$$

Consider that the starting of the highly overlapped linguistic variables might not be in sequential presentation as in Eqs. 7-9 which means that the starting and ending of each linguistic variable can occur arbitrarily, let us define a step function $u(x-a_i)$ to eliminate the contribution part of the linguistic variable in ΔS, when it still has not occurred yet or already exceeded: $u(x-a_i) = \begin{Bmatrix} 1 & x \geq a_i \\ 0 & x < a_i \end{Bmatrix}$

$\begin{Bmatrix} 0 & a_1 \leq x \leq a_j \\ 1 & x < a_1 \\ 1 & x > a_j \end{Bmatrix}$, So the general ΔS for both low and high overlapped linguistic variables can be calculated by Eq. 11.

The general laser simulator membership value for both low and highly overlapped linguistic variables can be calculated using Eq. 12

$$\Delta S = \sum_{i=1}^{j} \Delta S_i = \begin{Bmatrix} \min\big((x-a_1),(b_1-x)\big) + |(m_1-x)| \times \prod_{i=2}^{j}[1-u\,(x-a_i)] + \sum_{i=2}^{j} \mathrm{u}(\mathrm{x}-a_i) \times \min\big((x-a_i),(b_i-x)\big) & a_2 < x < b_i \\ \min\big((x-a_1),(b_1-x)\big) + |(m_1-x)| \times \prod_{i=3}^{j}[1-u\,(x-a_i)] + \sum_{i=3}^{j} \mathrm{u}(\mathrm{x}-a_i) \times \min\big((x-a_i),(b_i-x)\big) & b_1 < x < b_2 \\ \vdots & \\ \min\big((x-a_1),(b_1-x)\big) + |(m_1-x)| \times [1-u\,(x-a_i)] + \mathrm{u}(\mathrm{x}-a_j) \times \min\big((x-a_j),(b_j-x)\big) & b_{j-2} < x < b_{j-1} \end{Bmatrix} \quad (11)$$

$$\mu_i(x) = \begin{Bmatrix} 0 & x < a_i \vee x > a_i \\ \dfrac{\min\big((x-a_i),(b_i-x)\big)}{\min\big((x-a_i),(b_i-x)\big) + |(m_i-x)| \times [\prod_{i=i-1}^{i}[1-u\,(b_i-x)] + \prod_{i=i+1}^{j}[1-u\,(x-a_i)] + [\sum_{i=i-1}^{1} u(b_i-x) + \sum_{i=i+1}^{j} u(x-a_i)] \times \min\big((x-a_i),(b_i-x)\big)} & ;\ a_i \leq x \leq b_i \end{Bmatrix} \quad (12)$$

To eliminate the contribution part of the linguistic variable when it still has not occurred, we should use:

$u(x-a_i) \times \min\big((x-a_i),(b_i-x)\big) = \begin{Bmatrix} \min\big((x-a_i),(b_i-x)\big) & x \geq a_i \\ 0 & x < a_i \end{Bmatrix}$

To eliminate the contribution part of the linguistic variable when it has exceeded:

$u(b_i-x) \times \min\big((x-a_i),(b_i-x)\big) = \begin{Bmatrix} \min\big((x-a_i),(b_i-x)\big) & x \leq b_i \\ 0 & x > b_i \end{Bmatrix}$

Eq. 9 can be rewritten as in Eq. 10:

$$\Delta \boldsymbol{S} = \sum_{i=1}^{j} \Delta \boldsymbol{S_i} = \begin{Bmatrix} \mathbf{min}\,((\boldsymbol{x}-\boldsymbol{a_1}),(\boldsymbol{b_1}-\boldsymbol{x})) + \sum_{i=2}^{j} \mathbf{u}(\mathbf{x}-\boldsymbol{a_i}) \times \mathbf{min}\,((\boldsymbol{x}-\boldsymbol{a_i}),(\boldsymbol{b_i}-\boldsymbol{x})) & \boldsymbol{a_2} < \boldsymbol{x} \leq \boldsymbol{b_1} \\ \mathbf{min}\big((\boldsymbol{x}-\boldsymbol{a_2}),(\boldsymbol{b_2}-\boldsymbol{x})\big) + \left[\sum_{i=1}^{2} \mathbf{u}(\mathbf{x}-\boldsymbol{a_i}) + \left(\sum_{i=3}^{j} \mathbf{u}(\mathbf{x}-\boldsymbol{a_i})\right] \times \mathbf{min}\,((\boldsymbol{x}-\boldsymbol{a_i}),(\boldsymbol{b_i}-\boldsymbol{x})\right) & \boldsymbol{b_1} < \boldsymbol{x} \leq \boldsymbol{b_2} \\ \vdots & \\ \mathbf{min}\big((\boldsymbol{x}-\boldsymbol{a_{j-1}}),(\boldsymbol{b_{j-1}}-\boldsymbol{x})\big) + \left[\sum_{i=i-1}^{i} \mathbf{u}(\mathbf{x}-\boldsymbol{a_i}) + \left(\sum_{i=i+1}^{j} \mathbf{u}(\mathbf{x}-\boldsymbol{a_i})\right] \times \mathbf{min}\,((\boldsymbol{x}-\boldsymbol{a_i}),(\boldsymbol{b_i}-\boldsymbol{x})\right) & \boldsymbol{b_{j-2}} < \boldsymbol{x} \leq \boldsymbol{b_{j-1}} \end{Bmatrix} \quad (10)$$

To establish a general equation that can determine the membership values in both low and high overlapping cases, there is a need to include the part $|(m_i-x)|$ in Eq. 10 when there is a low overlapping, and this part should be neglected when there is a high occurrence of overlapping.

This can be done by using step function $u(x-a_i)$ in the form: $1-u(x-a_i) = \begin{Bmatrix} 0 & x \geq a_i \\ 1 & x < a_i \end{Bmatrix}$, which will be inserted into Eq. 10 using multiplication form as follows:

$\prod_{i=1}^{j}[1-u(x-a_i)] = \big(1-u(x-a_1)\big) \times \big(1-u(x-a_2)\big) \times \ldots \times \big(1-u\big(x-a_j\big)\big) =$

The general accumulative values of membership function for both low and high overlapping based on ΔS are calculated in Eq. 11, is as:

$$\sum_{i=1}^{j} \mu_i\,(x) = \begin{Bmatrix} 0 & x \leq a_1 \\ \frac{\min\big((x-a_1),(b_1-x)\big)}{(x-a_1)+|(m_1-x)|} & a_1 < x \leq a_2 \\ \sum_{i=1}^{j} \frac{\min\big((x-a_i),(b_i-x)\big)}{\Delta S} & a_2 < x \leq b_1 \\ \sum_{i=2}^{j} \frac{\min\big((x-a_i),(b_i-x)\big)}{\Delta S} & b_1 < x \leq b_2 \\ \vdots & \\ \sum_{i=j-1}^{j} \frac{\min\big((x-a_i),(b_i-x)\big)}{\Delta S} & b_{j-2} < x \leq b_{j-1} \\ \frac{\min\big((x-a_j),(b_j-x)\big)}{(b_j-x)+|(x-m_j)|} & b_{j-1} < x \leq b_j \\ 0 & x \geq b_j \end{Bmatrix} \quad (13)$$

5. Mathematical and Analytical Comparison between Fuzzy Logic and Laser Simulator Logic

To prove that Laser Simulator logic performs better than fuzzy logic when they are used in a noisy environment with highly overlapped linguistic variables, mathematical and analytical comparisons are provided as follows:

5.1 Mathematical Comparison

To mathematically prove that fuzzy logic is no longer suitable for a high degree of overlapping between linguistic variables, let us assume that we have an input fuzzy set with triangular membership functions, in which several linguistic variables *j* are highly overlapped with each other in following parameters: $a_1, a_2, a_3, \ldots\ldots, a_j$ are the starting points of the overlapped

linguistic variables, b_1,b_2, b_3,……, b_j are the end points of the overlapped linguistic variables and m_1, m_2, m_3, ……., m_j are the middle points of the overlapped linguistic variables. We have chosen triangular membership function due to ~~of~~ its similarity to a laser simulator in lowly overlapped linguistic variables. The membership function for each linguistic variable can be determined by Eq. 14:

$$\mu_i(x) = \begin{cases} 0 & x < a_i \\ 1 - \frac{2|x-m_i|}{b_i-a_i} & a_i \le x \le b_i \\ 0 & x > b_i \end{cases} \quad (14)$$

If the point is located in the highly overlapped area, then its accumulative membership value is calculated as in Eq. 15:

$$\sum_{i=1}^{j} \mu_i(x) = \begin{cases} 0 & x < a_1 \\ 1 - \frac{2|x-m_1|}{b_1-a_1} & a_1 \le x \le a_2 \\ 1 - \frac{2|x-m_1|}{b_1-a_1} + 1 - \frac{2|x-m_2|}{b_2-a_2} + \cdots + 1 - \frac{2|x-m_j|}{b_j-a_j} & a_2 < x \le b_1 \\ 1 - \frac{2|x-m_2|}{b_2-a_2} + 1 - \frac{2|x-m_3|}{b_3-a_3} + \cdots + 1 - \frac{2|x-m_j|}{b_j-a_j} & b_1 < x \le b_2 \\ \vdots & \\ 1 - \frac{2|x-m_j|}{b_j-a_j} & b_{j-1} < x \le b_j \\ 0 & x > b_j \end{cases} \quad (15)$$

To find out the point of membership function, in which the accumulative membership values starts to become higher than 1, let us assume the membership functions with two highly overlapped linguistic variables. More highly overlapped linguistic variables will result in a high probability to get membership values higher than 1. Eq.7 can be re-written with two linguistic variables as in Eq. 8:

$$\mu_1(x) + \mu_2(x) = \begin{cases} 1 - \frac{2|x-m_1|}{b_1-a_1} & x < a_2 \\ 2 - \frac{2|x-m_1|(b_2-a_2)+2|x-m_2|(b_1-a_1)}{(b_1-a_1)(b_2-a_2)} & a_2 \le x \le b_1 \\ 1 - \frac{2|x-m_2|}{b_2-a_2} & x > b_2 \end{cases} \quad (16)$$

If $m_1 = a_1 + \frac{b_1-a_1}{2}$ and $m_2 = a_2 + \frac{b_2-a_2}{2}$, one can write Eq. 16 as follows:

$$\mu_1(x) + \mu_2(x) = \begin{cases} 1 - \frac{|2x-a_1-b_1|}{b_1-a_1} & a_1 \le x \le a_2 \\ 2 - \frac{2|x|(b_2-a_2)+2|x|(b_1-a_1)+a_1a_2-b_1b_2}{(b_1-a_1)(b_2-a_2)} & a_2 < x \le b_1 \\ 1 - \frac{|2x-a_2-b_2|}{b_2-a_2} & b_1 < x \le b_2 \end{cases} \quad (17)$$

By solving Eq. 17, one can determine the values of x in the highly overlapped area $a_2 \le x \le b_1$ that are bigger than 1 as follows:

$$\mu_1(x) + \mu_2(x) = 2 - \frac{2|x|(b_2-a_2)+2|x|(b_2-a_2)+a_1a_2-b_1b_2}{(b_i-a_i)(b_2-a_2)} > 1$$

The solutions are:

$$\mu_1(x) + \mu_2(x) \le 1 \text{ if } x \le \frac{a_1a_2+a_1b_2+a_2b_1-3b_1b_2}{2(a_1+a_2-b_1-b)} \text{ or } x \ge b_i - \frac{a_1a_2+a_1b_2+a_2b_1-3b_1b_2}{2(a_1+a_2-b_1-b)};$$

However $\mu_1(x) + \mu_2(x) \ge 1$ if $\frac{a_1a_2+a_1b_2+a_2b_1-3b_1b_2}{2(a_1+a_2-b_1-b)} \le x \le b_i - \frac{a_1a_2+a_1b_2+a_2b_1-3b_1b_2}{2(a_1+a_2-b_1-b)}$

An example is $x=m_i$ which is located in a range $a_{i+1} \le x \le b_i$, Equation 3.16 can be written:

$$\mu_1(x) + \mu_2(x) = 2 - \frac{2|m_1-m_2|(b_1-a_1)}{(b_1-a_1)(b_2-a_2)} = 2 - \frac{2|m_1-m_2|}{(b_2-a_2)} = 2 - \frac{|(a_1-a_2)+(b_1-b_2)|}{(b_2-a_2)}$$

Since there is an overlap between the linguistic variables, this expression will always be guaranteed: $\frac{|(a_1-a_2)+(b_1-b_2)|}{(b_2-a_2)} < 1$ and thus $\mu_1(x) + \mu_2(x) > 1$.

The border $\frac{|(a_1-a_2)+(b_1-b_2)|}{(b_{i+1}-a_{i+1})}$ becomes equal or bigger than 1, only if there is no overlapping between linguistic variables *i.e.*, the latter's linguistic starts from the middle (or after the middle) of the former linguistic variables and the former linguistic variable ends at the middle of the latter linguistic variable (or before the middle). The same goes to $x=m_2$

$$\mu_1(x) + \mu_2(x) = 2 - \frac{2|m_2-m_1|(b_2-a_2)}{(b_i-a_i)(b_2-a_2)} = 2 - \frac{2|m_2-m_1|}{(b_1-a_1)} = 2 - \frac{|(a_2-a_1)+(b_2-b_1)|}{(b_1-a_1)} \ge 1$$

This will be resulted by errors when implicating the membership values of highly overlapped linguistic variables of fuzzy input memberships on the output membership functions based on the consequence of the rules.

Eq. 18 and 19 show~~s~~ the crisp output after implication and defuzzification using centroid in Mamdani and singleton in Sugeno, ~~in~~ respectively.

$$y = \frac{\int \mu_i(x) y dy}{\mu_i(x) dy} \approx \frac{\sum_{i=1}^{n} \mu_i(x) \times \Delta y_i}{\sum_{i=1}^{n} \mu_i(x)} \quad (18)$$

$$y = \frac{\sum_{i=1}^{n} \mu_i(x) \times y_i}{\sum_{i=1}^{n} \mu_i(x)} \quad (19)$$

where n is the number of output linguistic variables as stated in the rules. Δy_i and y_i are the universes of discourses of output membership that have been implicated by the membership value of input membership in Mamdani and Sugeno, respectively. Since the accumulative membership~~s~~ values are highly overlapped, linguistic variables that are bigger than 1 in certain ranges, $\mu_i(x)$ can be decomposed into:

$$\mu_i(x) = \mu_{ti}(x) + \delta_i \quad (20)$$

where $\mu_{ti}(x)$ are the right membership values when the accumulative membership values of highly overlapped linguistic variables become less or equal to 1. δ_i are the errors raised by the effect of highly overlapped linguistic variables. Eq. 18 and 19 can be rewritten as Eq. 21 and 22, respectively:

$$y = \frac{\sum_{i=1}^{n} \mu_i(x) \times \Delta y_i}{\sum_{i=1}^{n} \mu_i(x)} = \frac{\sum_{i=1}^{n} (\mu_{ti}(x)+\delta_i) \times \Delta y_i}{\sum_{i=1}^{n} (\mu_{ti}(x)+\delta_i)} = \frac{\sum_{i=1}^{n} \mu_{ti}(x) \times \Delta y_i}{\sum_{i=1}^{n} (\mu_{ti}(x)+\delta_i)} + \frac{\sum_{i=1}^{n} \delta_i \times \Delta y_i}{\sum_{i=1}^{n} (\mu t_i(x)+\delta_i)} \quad (21)$$

$$y = \frac{\sum_{i=1}^{n} \mu_i(x) \times y_i}{\sum_{i=1}^{n} \mu_i(x)} = \frac{\sum_{i=1}^{n} (\mu_{ti}(x)+\delta_i) \times y_i}{\sum_{i=1}^{n} (\mu_{ti}(x)+\delta_i)} = \frac{\sum_{i=1}^{n} \mu_{ti}(x) \times y_i}{\sum_{i=1}^{n} (\mu_{ti}(x)+\delta_i)} + \frac{\sum_{i=1}^{n} \delta_i \times y_i}{\sum_{i=1}^{n} (\mu t_i(x)+\delta_i)} \quad (22)$$

Where the expression $\frac{\sum_{i=1}^{n} \delta_i \times x_i}{\sum_{i=1}^{n} (\mu_{ti}(x)+\delta_i)}$ indicates the errors caused by the highly overlapped linguistic variables.

If the laser simulator is used to find out the membership values of the two highly overlapped linguistic variables as in Eq. 17, it can be written as similar to Eq. 23

$$\mu_1(x)+\mu_2(x)=\begin{cases}0 & x\le a_1\\ \frac{x-a_1}{(x-a_1)+(m_1-x)} & a_1<x\le a_2\\ \frac{x-a_1}{(x-a_1)+(x-a_2)}+\frac{x-a_2}{(x-a_1)+(x-a_2)} & a_2<x\le m_1\\ \frac{b_1-x}{(b_1-x)+(x-a_2)}+\frac{x-a_2}{(b_1-x)+(x-a_2)} & m_1<x\le m_2\\ \frac{b_1-x}{(b_1-x)+(x-a_2)}+\frac{b_2-x}{(b_1-x)+(x-a_2)} & m_2<x\le b_1\\ \frac{b_2-x}{(b_2-x)+(m_2-x)} & b_1<x\le b_2\\ 0 & x>b_2\end{cases} \quad (23)$$

From Eq. 23, it is clearly stated that the accumulative values of highly overlapped areas ($a_2<x\le m_1, m_1<x\le m_2$ and $m_2<x\le b_1$) can't exceed 1. Thus the implication of fuzzy input memberships on the output membership functions based on consequent of the rules, will be accomplished using Eq. 4 without errors.

$$y=\frac{\sum_{i=1}^{n}\mu_i(x)\times Y_i}{n} \quad (24)$$

where Y_i is the output linguistic range.

Thus the difference between crisp outputs in fuzzy logic y_{FL} and laser simulator y_{LS} for highly overlapped linguistic variables can be written:

$$y_{FL}-y_{LS}=\frac{\sum_{i=1}^{n}\delta_i\times x_i}{\sum_{i=1}^{n}(\mu_i(x))} \quad (25)$$

5.2 Analytical comparison between Fuzzy logic and Laser Simulator

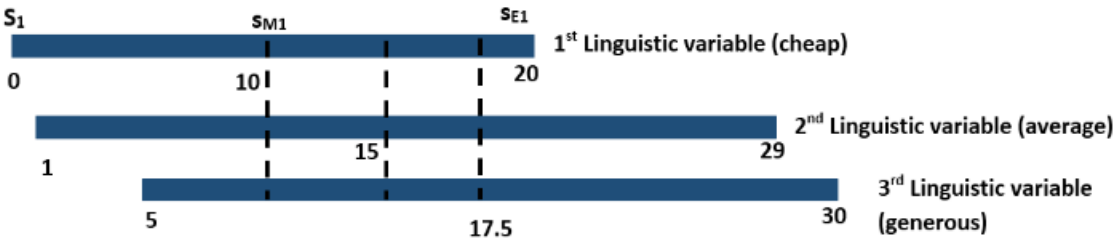
From Eqs. 1 and 6, it is noticeable that the membership function values in laser simulator logic are similar to symmetric triangular membership function of fuzzy logic type-1 in the case where there is no/low overlapping between linguistic variables. However it works differently when there is an overlapping between the linguistic variables such as in Fig. 8, where the accumulative membership values of fuzzy logic exceed 1. Fig. 8 shows a fuzzy set (tips) with three linguistic variables with a degree of overlapping (cheap 0-20, average 1-29, generous 5-30). The laser simulator logic has the capability to deal well with a high degree of inference/overlapping between linguistic variables as shown in Fig. 8.a. In fact, the inference between linguistic variables in the current fuzzy systems is determined arbitrary and the universe of discourse is just divided into equal partitions as shown in Fig. 4. Another common practice in the current fuzzy logic type 1 or 2 algorithms is that one can't overlap more than two linguistic variables as the total membership value in this case becomes bigger than 1 as shown in Fig. 8.b. The laser simulator can solve this problem by generating a dynamic membership function in which one can overlap easily more than two linguistic variables as shown in Fig. 8.a.

Based on fuzzy logic type 1 and 2, the input x=10 has a total membership value is approximately $\mu=1.9$ as shown in Fig. 8.b which is not valid, since the accumulative membership values of a crisp input should not exceed 1. However, the laser simulator assures that the membership value of each input value can't exceed the value of 1 using Eq. 1 and 2.

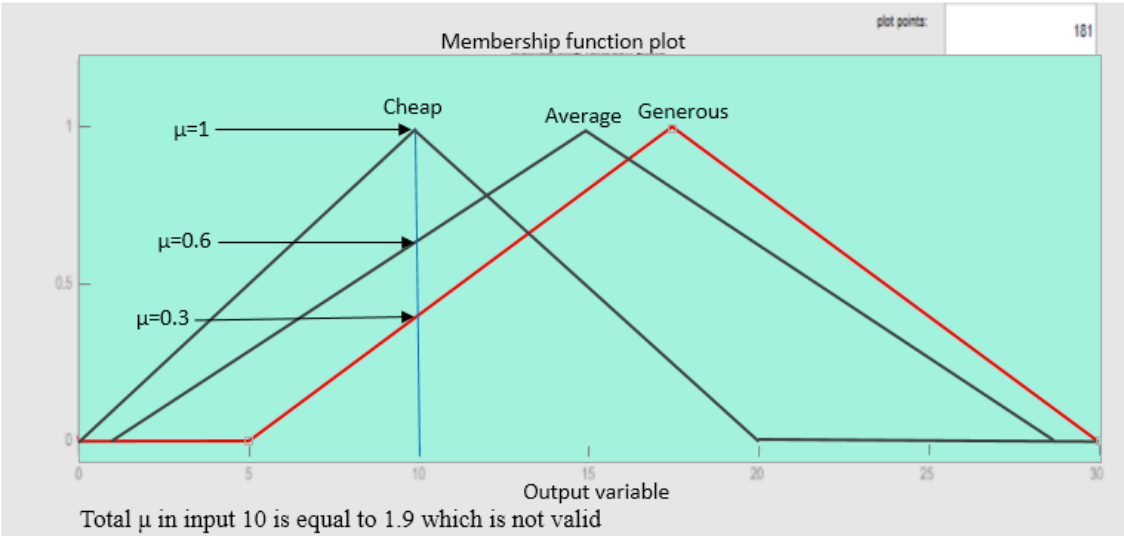
The calculation when crisp input *x=10* for laser simulator is accomplished based on Eq. 8:

For the 1st linguistic variable (cheap):

$$\mu_1=\frac{x-L_{S1}}{(x-L_{S1})+(x-L_{S2})+(x-L_{S3})}=\frac{10-0}{(10-0)+(10-1)+(10-5)}=\frac{10}{24}$$



(a) Laser Simulator Logic



(b) Fuzzy Logic

Fig. 8 Comparison between laser simulator logic and fuzzy logic when there is an overlapping

For the 2nd linguistic variable (average):

$$\mu_2 = \frac{x - L_{S2}}{(x - L_{S1}) + (x - L_{S2}) + (x - L_{S3})} = \frac{10 - 1}{(10 - 0) + (10 - 1) + (10 - 5)} = \frac{9}{24}$$

For the 3rd linguistic variable (generous):

$$\mu_3 = \frac{x - L_{S3}}{(x - L_{S1}) + (x - L_{S2}) + (x - L_{S3})} = \frac{10 - 5}{(10 - 0) + (10 - 1) + (10 - 5)} = \frac{5}{24}$$

So $\mu = \mu_1 + \mu_2 + \mu_3 = \frac{24}{24}$

As a result, the capability of noise elimination and simplification of drawing in laser simulator logic help to easily find the start, end and range of linguistic variables if we know some samples of input/output. Table 3 summarizes the difference between laser simulator logic and fuzzy logic

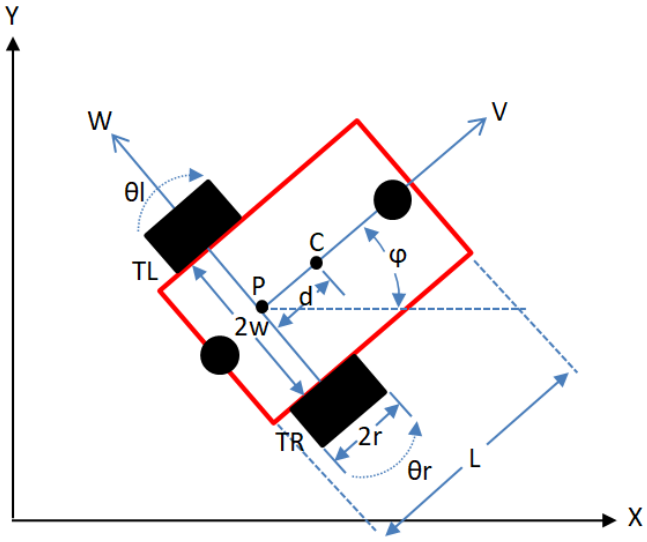


Fig. 9: Local /global coordinate system with wheeled mobile robot

Where
C: The center of mass mobile robot, d: The distance between the center of mass and driving wheels axis in the x-direction

Table 3 Laser Simulator Logic vs Fuzzy Logic

Features	**Fuzzy logic**	**Laser Simulator Logic**
Accumulative membership function	Not guaranteed to be less or equal to 1	Guaranteed to be less or equal to 1
Calculation of implication/ crisp output	The calculation is hard and rare to be used. Thus it depends mainly on the drawings of input/output membership function to find the implication.	The calculation is simple and can be easily implemented as discussed in Section 6.2.
Dealing with Highly overlapping of linguistic variables	Fuzzy logic becomes noisy in this case	It works well with high noisy
Drawing	Hard to be drawn manually	Simple
Ability for learning	Can't learn directly if we have a sample of input/output. Thus we use an additional algorithm for learning such as genetic, neural networks ,etc.	Can be leant if we have some samples of input and output.
Real time systems	It needs a lot of experiments to adjust the inference of membership functions	Inference system can be easily adjusted especially if we have samples of input/output
Nature of fuzzification	Can be linear (triangular and trapezoidal membership) or non-linear (Gaussian)	Linear. The crisp value has proportional ratio to its position in the range of membership function linguistic variable
Nature of defizzification	Several methods can be used: center of gravity, adaptive integration, etc.	Weighted average

6. Modelling of Wheeled Mobile Robot

The wheeled mobile robot in this research is driven by two differential and two castor wheels to enable the robot to move in the constrained environments using active force control (AFC) as shown in Fig.9. The local and global coordinate systems are used to describe the movement of the wheeled mobile robot as shown in Fig. 9.

P: The intersection of the axis of symmetry with the driving wheels axis
w: The distance between each driving wheel and the robot axis of symmetry in the y-direction

6.1 Kinematics of Wheeled Mobile Robot

The wheeled mobile robot is driven by two differential drive and two castor wheels.

The velocity of the wheeled mobile robot on the right and left wheels can be calculated using Eq. 26:
Velocity for right wheel is*: Vr* = $\dot{\theta}_r .r$
Velocity for left wheel is*: V_l* = $\dot{\theta}_l .r$
Where θ_r and θ_l are the angular velocity of the right and left wheel, respectively.

$$V = \frac{Vr + Vl}{2} = \frac{r\dot{\theta}_r + r\dot{\theta}_l}{2} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \quad (26)$$

The heading rotation angle of the robot can then be calculated as the difference between the angular velocity of the right and left wheels as in Eq. 27:

$$\dot{\varphi} = \frac{r\dot{\theta}_r - r\dot{\theta}_l}{2w} = \begin{bmatrix} \frac{r}{2w} & \frac{-r}{2w} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \quad (27)$$

The global coordinate system (x, y, φ) can be described by means of the local coordinate system (V,ω) as shown in Eq. 28

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} cos\varphi & dsin\varphi \\ sin\varphi & -dcos\varphi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ \dot{\varphi} \end{bmatrix} \quad (28)$$

The velocity of the robot in x and y directions ($\dot{x}$ *and* $\dot{y}$) can be written by means of the wheel's angular velocity as described in Eqs. 29 and 30:

$$\dot{x} = \begin{bmatrix} \frac{r\cos\varphi}{2} & \frac{r\cos\varphi}{2} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} + \begin{bmatrix} \frac{r\mathrm{d}\sin\varphi}{2w} & \frac{-r\mathrm{d}\sin\varphi}{2w} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \quad (29)$$

$$\dot{y} = \begin{bmatrix} \frac{rsin\varphi}{2} & \frac{rsin\varphi}{2} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} - \begin{bmatrix} \frac{drcos\varphi}{2w} & \frac{-drcos\varphi}{2w} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \quad (30)$$

Based on Eqs. 26-30, the velocity of wheel mobile robot in x, y and φ directions can be written as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\varphi + \frac{dr}{2w}\sin\varphi & \frac{r}{2}\cos\varphi - \frac{dr}{2w}\sin\varphi \\ \frac{r}{2}sin\varphi - \frac{dr}{2w}cos\varphi & \frac{r}{2}sin\varphi + \frac{dr}{2w}cos\varphi \\ \frac{r}{2w} & \frac{-r}{2w} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \quad (31)$$

6. 2 Dynamic of Mobile Robot

The dynamic of a mobile robot is calculated using the Lagrange equation. In fact, this equation depends on the calculation of the potential and kinetic energy of the whole system. It considers that the energy inside the system is equal to the energy applied by external forces as in Eq. 32. The Lagrange dynamic equation can be written as Eq. 32:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) + \frac{\partial L}{\partial q} = \tau - A^T(q)\lambda \quad (32)$$

Where, $L = K - V$
K= kinetic energy,
V = potential energy in system,
τ = torque vector
A^T (q) = transpose of constraint matrix
λ = lagrange multipliers vector.
Since robot height is just 20-30cm, the potential energy is too small and therefore can be neglected, so:
$L = K - V = K$
To establish the kinetic energy, the robot has been divided into three parts, namely the robot's body, left and right wheels.

Robot body

The kinetic energy of the body is raised from the mass-energy of its elements during movements such as aluminium sheets and profiles, electrical battery and cards, motor and gearbox. The kinetic energy of the body contains translational and rotational kinetic energies as follows:
So, the total energy for the body is:

$$K_b = \frac{1}{2} m_b \dot{x}^2 + \frac{1}{2} m_b \dot{y}^2 + \frac{1}{2} I_b \dot{\varphi}^2 \quad (33)$$

The total kinetic energy on the right wheel is:

$$K_{rw} = \frac{1}{2} m_{wR} (\dot{x} + b\dot{\varphi}\cos\varphi - d\dot{\varphi}\sin\varphi)^2 + \frac{1}{2} m_{wR} (\dot{y} + b\dot{\varphi}\sin\varphi + d\dot{\varphi}\cos\varphi)^2 + \frac{1}{2} I_{wR} \dot{\theta}_R^{\,2} + \frac{1}{2} I_{wR} \dot{\varphi}^2 \quad (34)$$

The total Kinetic energy on the left wheel is:

$$K_{lw} = \frac{1}{2} m_{wL} (\dot{x} + b\dot{\varphi}\cos\varphi + d\dot{\varphi}\sin\varphi)^2 + \frac{1}{2} m_{wL} (\dot{y} + b\dot{\varphi}\sin\varphi + d\dot{\varphi}\cos\varphi)^2 + \frac{1}{2} I_{wR} \dot{\theta}_R^{\,2} - \frac{1}{2} I_{wR} \dot{\varphi}^2 \quad (35)$$

The total kinetic energy for the wheeled mobile robot can be written as:

$L = K_b + K_{wr} + K_{wl}$

$$L = \frac{1}{2} m_b \dot{x}^2 + \frac{1}{2} m_b \dot{y}^2 + \frac{1}{2} I_b \dot{\varphi}^2 + \frac{1}{2} m_{wR} (\dot{x} + b\dot{\varphi}\cos\varphi - d\dot{\varphi}\sin\varphi)^2 + \frac{1}{2} m_{wR} (\dot{y} + b\dot{\varphi}\sin\varphi + d\dot{\varphi}\cos\varphi)^2 + \frac{1}{2} I_{wR} \dot{\theta}_R^{\,2} + \frac{1}{2} I_{wR} \dot{\varphi}^2 + \frac{1}{2} m_{wL} (\dot{x} - b\dot{\varphi}\cos\varphi + d\dot{\varphi}\sin\varphi)^2 + \frac{1}{2} m_{wL} (\dot{y} + b\dot{\varphi}\sin\varphi + d\dot{\varphi}\cos\varphi)^2 + \frac{1}{2} I_{wL} \dot{\theta}_L^{\,2} - \frac{1}{2} I_{wR} \dot{\varphi}^2 \quad (36)$$

By deriving L $\frac{\mathbf{dL}}{\boldsymbol{d\dot{q}}}$, $\frac{\boldsymbol{d}}{\boldsymbol{dt}} . \frac{\mathbf{dL}}{\boldsymbol{d\dot{q}}}$ and $\frac{\mathbf{dL}}{\boldsymbol{dq}}$ Eq. 36 can be written as Eq. 40.
Since the wheeled mobile robot has a non-holonomic movement, the following constraints are applied to the robot movement:

1- No slippage during movement
2- There are only forward and backward movements; no movements in the side of the wheels

These constraints can be written in Eqs. 37-39 as follows:

$$m\ddot{x} + 2m_w \mathrm{d}\sin\varphi\,\ddot{\varphi} + 2m_w \mathrm{d}\cos\varphi\,\dot{\varphi}^2 = 0 \quad (37)$$

$$m\ddot{y} - 2m_w \mathrm{d}\cos\varphi\,\ddot{\varphi} + 2m_w \mathrm{d}\,\dot{\varphi}^2 \sin\varphi = 0 \quad (38)$$

$$I\ddot{\varphi} + + 2m_w \mathrm{d}\sin\varphi\,\ddot{x} - 2m_w \mathrm{d}\cos\varphi\,\ddot{y} = 0 \quad (39)$$

So the general equation to describe the dynamic movement of the robot can be written as:

$$M(q)\ddot{q} + V(q,\dot{q}) = \tau - A^T(q)\lambda \quad (40)$$

$$\begin{bmatrix} m & 0 & 2m_\omega \mathrm{d}\sin\varphi & 0 & 0 \\ 0 & m & -2m_\omega \mathrm{d}\sin\varphi & 0 & 0 \\ 2m_\omega \mathrm{d}\sin\varphi & -2m_\omega \mathrm{d}\sin\varphi & I & 0 & 0 \\ 0 & 0 & 0 & I_w R & 0 \\ 0 & 0 & 0 & 0 & I_w L \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\varphi} \\ \ddot{\theta}R \\ \ddot{\theta}L \end{bmatrix} + \begin{bmatrix} 2m_w \mathrm{d}\cos\varphi\,\dot{\varphi}^2 \\ 2m_w \mathrm{d}\,\dot{\varphi}^2 \sin\varphi \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \tau_r \\ \tau_l \end{bmatrix} -$$

$$+ \begin{bmatrix} -\sin(\varphi) & \cos(\varphi) & -d & 0 & 0 \\ \cos(\varphi) & \sin(\varphi) & -b & -r & 0 \\ \cos(\varphi) & \sin(\varphi) & b & -r & 0 \end{bmatrix}^T \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}$$

Where m is the total mass of the mobile robot which is m = $m_b + m_{wR} + m_{wL}$, τ_r and τ_l are the applied torques on the right and left wheels, respectively. *m* is the total mass of robot, m_w: is mass of the wheel, *I* is the moment inertia of the robot and I_w is the moment inertia of the wheel.
Eq. 40 can be re-written if one replaces $\ddot{x}$, $\ddot{y}$ and $\ddot{\varphi}$ from Eq.31 (after applying second derivation) with $S(\varphi) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix}$ as Eq. 41:

$$\mathbf{M}(q)S(\varphi)\ddot{\theta}+\mathbf{M}(q)\dot{S}(\varphi)\dot{\theta}+\mathbf{V}(\dot{q})=\tau-A^{\mathrm{T}}(q)\lambda \qquad (41)$$

If Eq. 41 is multiplied by $S^{\mathrm{T}}(\varphi)$ (transpose of $S(\varphi)$), the part $S^{\mathrm{T}}(\varphi)A^{\mathrm{T}}(q)$ becomes equal to zero as stated in Eqs. 37-39. Thus the final equation of WMR dynamics can be written as Eq. 42:

$$S^{\mathrm{T}}(\varphi)\mathbf{M}(q)S(\varphi)\ddot{\theta}+S^{\mathrm{T}}(\varphi)\mathbf{M}(q)\dot{S}(\varphi)\dot{\theta}+S^{\mathrm{T}}(\varphi)V(\dot{q})=S^{\mathrm{T}}(\varphi)\tau \qquad (42)$$

6.3 Design of Controller

The proposed PD-AFC-LS-QC controller consists of three feedback control loops as shown in Fig. 10, namely the internal, external and quick compensation loops to effectively compensate the disturbances in the constrained environments. The external loop is used to control the kinematics parameters of the WMR control system via a PD controller. However, the internal loop is used to control the dynamic of robot and disturbance rejection via AFC controller. On the other hand, a quick compensation loop has been introduced in this paper to compensate the difference between the reference and actual acceleration via the P controller, especially at the beginning of a movement. Thus, the proposed PD-AFC-LS-QC controller will possess the advantages of PD, AFC and QC controllers.
In the proposed controller, the PD is used to control the kinematics parameters of the mobile robot through a continuous comparison between the reference and actual acceleration in x, y and φ as in Eqs. 43-45

$$\ddot{x}_{act} = \ddot{x}_{ref} + K_d(\dot{x}_{ref} - \dot{x}_{act}) + K_p(x_{ref} - x_{act}) \qquad (43)$$

$$\ddot{y}_{act} = \ddot{y}_{ref} + K_d(\dot{y}_{ref} - \dot{y}_{act}) + K_p(y_{ref} - y_{act}) \qquad (44)$$

$$\ddot{\varphi}_{act} = \ddot{\varphi}_{ref} + K_d(\dot{\varphi}_{ref} - \dot{\varphi}_{act}) + K_p(\varphi_{ref} - \varphi_{act}) \qquad (45)$$

Whereas, AFC is used to estimate the disturbances affected to the control system by comparing the applied torque on actuators (motor) with the actual torque measured by sensors as in Eq. 46

$$\tau_d^* = \tau - IN\,\ddot{\theta} \qquad (46)$$

where *IN* is inertia matrix, $\ddot{\theta}$ is wheel's angular velocity, τ_d^* is the disturbance torque applied on the wheel and τ is the DC-motor's torque. The DC-motor has a linear relationship between the torque and current: $\tau = K_t I_t$, where I_t is the current supplied to DC-motor and K_t is the DC-motor torque's constant. The moment's inertia IN can't be measured using sensors. Thus, it is estimated through a well-known AI algorithms such as crude-approximations (Hewit and Burdess, 1981), iterative-learning (Mailah, 1998), neural-network (Kwek et al., 2003), knowledge-based expert (Pitowarno et al., 2003) and fuzzy-logic (FL) (Ali and Mailah, 2019a, 2019b).

As explained in Section 4, a novel method, called laser simulator logic has been developed to estimate the inertia moment of WMR during movement. The moment of inertia will then be multiplied by the angular velocity of the robot wheel to estimate the actual torque.
The added QC loop enables a quick compensation of the acceleration, especially at the begining of the movement , where PD faces some delays, due to AFC loop calculations and AI-based Inertia moment estimation. The compensation is calculated using Eq. 47

$$e = kp(\ddot{x}_{act} - \ddot{x}_{ref}) \qquad (47)$$

6.4 Estimation of Inertia Matrix

The mass matrix **M**(q) in Eq. 34 is used to determine the range of the inertia moment values to overcome the disturbances. It has been found practically by Mailah, 1998 that the range of inertia moment can be described as in Eq. 48:

$$0.4\mathbf{M}(q) \le IN \le 1.4\mathbf{M}(q) \qquad (48)$$

The designs of fuzzy logic and laser simulator logic algorithms are passed through three main processes, namely; fuzzification, rule interface and defuzzification to estimate the

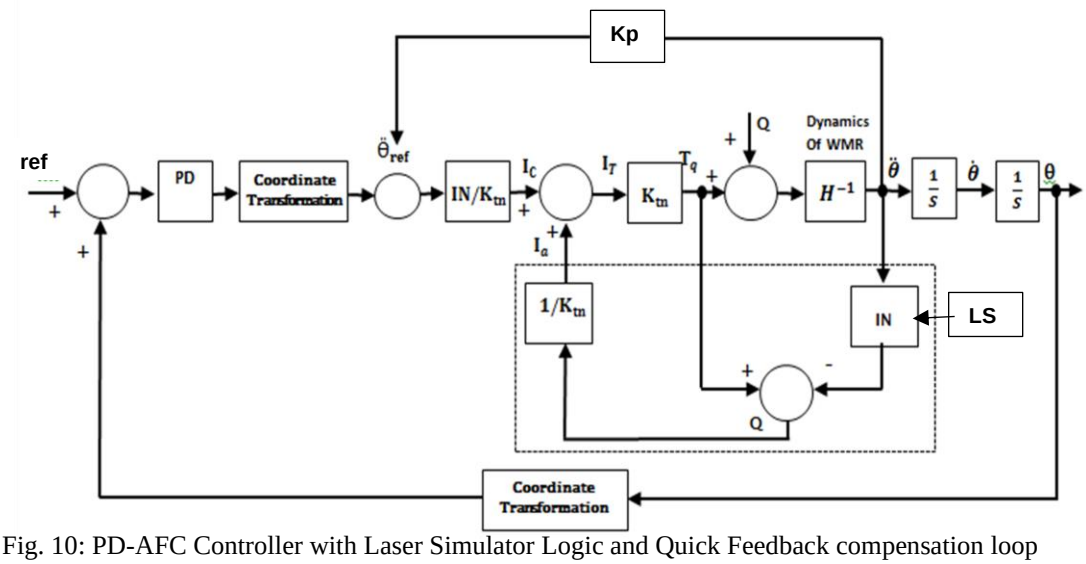


Fig. 10: PD-AFC Controller with Laser Simulator Logic and Quick Feedback compensation loop

inertia matrix based on the changes of head angle of the robot φ. Heading angle φ has been chosen as the input set of the fuzzy logic and laser simulator logic algorithms since the inertia matrix M(q) varies only based on its values along with the time as illustrated in Eq. 40. Two output fuzzy sets are used for calculating the wheel's inertia matrix, namely the right and left wheel inertia matrices. The membership functions for both fuzzy logic and laser simulator have highly overlapped linguistic variables, since it the control system is so noisy in this work.

Fuzzification: With this process, the crisp values are transformed into linguistic variables, which are involved in the input or output fuzzy sets. The input fuzzy set (φ) is fuzzified into the following linguistic variables:

φ = {Very Low, Low, Medium, High, Very High}.

Two output-fuzzy sets are used to calculate the wheel's inertia matrix, namely the right and left wheel inertia matrices which are decomposed into the following linguistic variables:

INR-INL = {Very Small, Small, Medium, Large, Very Large}.

The triangular membership function has been chosen to present the input/output sets of the fuzzy logic algorithm due to of the similarity between the fuzzy logic with triangular function and the laser simulator which makes it easy for comparison later on. The overlapping between the linguistic variables has been carried out after conducting some experiments, in which we found the following suitable ranges for each input/output linguistic variables as illustrated in Tables 4 and 5:

Table 4: input (φ ^{0}) linguistic variable ranges

Very Low	Low	Medium	High	Very High
0-70 ^{0}	20-160 ^{0}	80-250 ^{0}	150-300 ^{0}	280-360^0

Table 5: Output INR/INL (Kgm^2) linguistic variable ranges

Very Small	Small	Medium	Large	Very Large
0.335-0.360	0.345-0.395	0.365-0.415	0.398-0.435	0.418-0.440

It is clearly stated in Tables 3 and 4 that there are overlapping in the linguistic variables of input and output. The overlapping of linguistic variables is considered same in both fuzzy and laser simulator logic to enable for analyzing the performance of each one as will be discussed in Section 7.

Fig. 11 illustrates the membership functions of the input/output fuzzy sets in fuzzy logic based on Tables 3 and 4. Fig. 12 shows the membership functions of the input/output sets in laser simulator logic based on Tables 3 and 4.

Rule Interface: Both of fuzzy logic and laser simulator logic uses the IF-THEN statements, which are utilized to describe the relationship between the input and output sets. In this work, the *Mamdani* method has been used for implication, aggregation of the rules in the fuzzy logic algorithm as follows: Rules in fuzzy logic:

1. If (phi is M) then (INL is ML) & (INR is MR)
2. If (phi is VL) then (INL is VLL) & (INR is VSR)
3. If (phi is L) then (INL is LL) & (INR is SR)
4. If (phi is H) then (INL is SL) & (INR is LR)
5. If (phi is VH) then (INL is VSL) & (INR is VLR)

Same rules, but with different style, are used in laser simulator as follows:

1 rule: If phi is Very Low Then INR is Very Small and INL is Very Large
2 rule: If phi is Low Then INR is Small and INL is Large
3 rule: If phi is Medium Then INR is Medium and INL is Medium
4. rule: If phi is High Then INR is Large and INL is Small
5. rule: If phi is Very High Then INR is Very Large and INL is Very Small

Defuzzification: Fuzzy logic uses the *Mamdani* inference mechanism for the defuzzification process with the *centroid weighted* method. The crisp output is given by Eq. 49:

$$x = \frac{\int \mu_i(x) x \, dx}{\mu_i(x) \, dx} \tag{49}$$

where $\mu_i(x)$ is membership value, *x* is a universe of discourse of output membership that has been implicated by the membership value of the input membership in Mamdani.

Laser simulator logic uses an average output in defuzzification process as follows:

$$x = \frac{\sum_{i=1}^{n} \mu_i(x) \times x_i}{n} \tag{50}$$

where *n* is the number of output linguistic variables as stated in the rules, Y_i is the output linguistic range.

7. Simulation Setup and Results

The simulation is performed in MATLAB/Simulink in two kinds of constrained paths, namely the zigzag and highly curverd paths, all with and without presence of disturbances. The performance of the proposed control system (PD-AFC with laser simulator and Quick feedback Compensation) PD-AFC-LS-QC has been compared with several control systems such as:

- PD controller (PD)
- AFC with Fuzzy logic (AFC-FL)
- AFC with laser simulator (AFC-LS)
- PD-AFC with fuzzy logic (PD-AFC-FL)
- PD-AFC with laser simulator (PD-AFC-LS)

7. 1 Simulation Results of Controllers without Disturbances

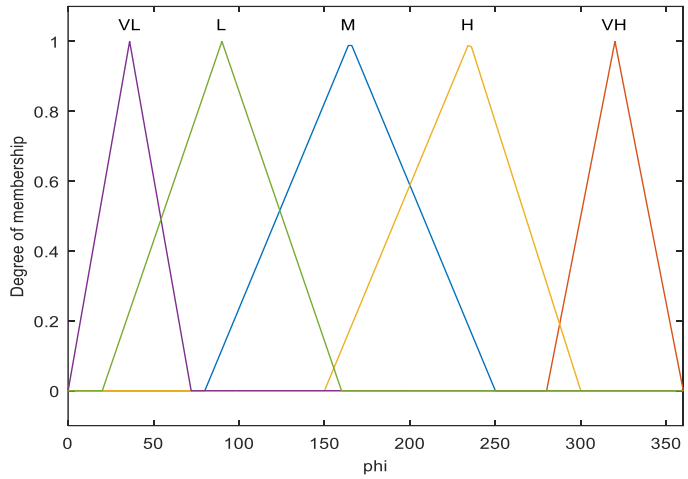
A zizag and circular paths were selected to test the new proposed control system PD-AFC-LS-QC and comparisons were made with other controllers. Fig. 13 shows the pre-planned path for testing the proposed control system.

The simulation has been run with the following parameters:
Simulation Parameters: Integration method: ODE3 (*Bogacki-Shanpine*)
Simulation step: Fixed step, fixed step size: Auto
WMR Parameters:

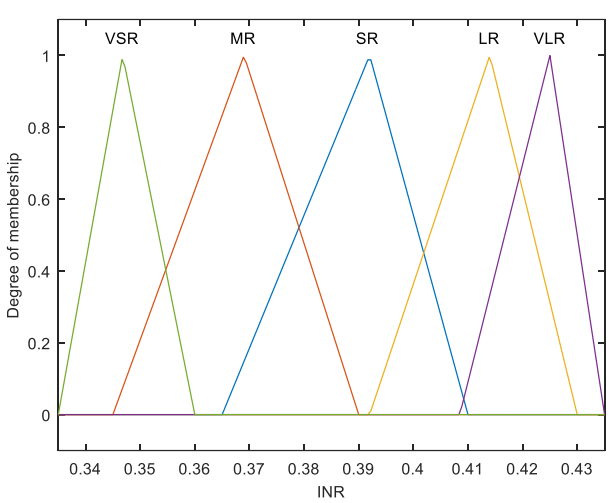
r = 0.15 m, b = 0.75 m, d = 0.03 m, m = 31.0 kg, m_w = 0.5 kg, I = 15.625 kgm^2 , I_w = 0.005 kgm^2

Controller Parameters:

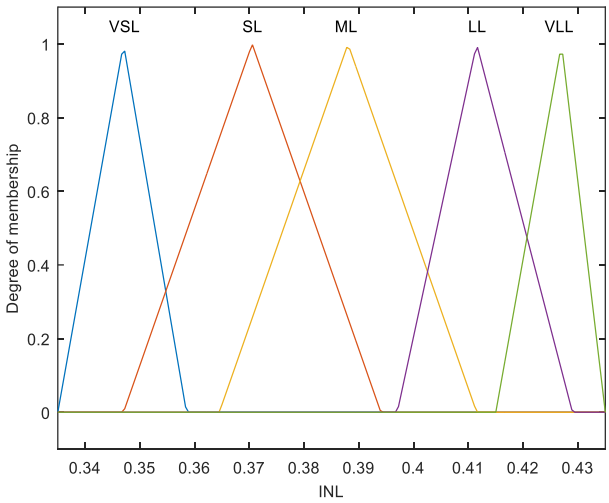
K_{px} = 1/s^2, K_{py} = 2/s^2, $K_{p\varphi}$= 1/s, K_{dx} = 1, K_{dy} =2 , $K_{d\varphi}$= 1, $K_t = [0.263 \;\; 0.263]^T$ Nm/A



(a) input (φ^0)

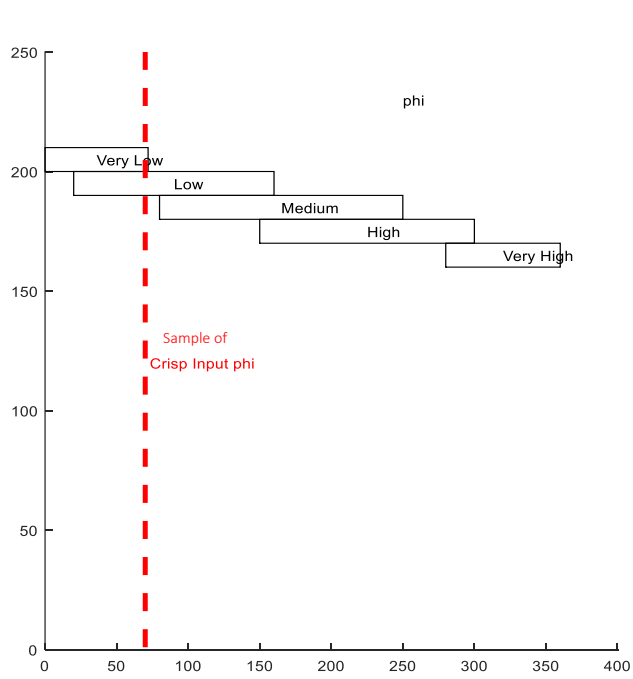


(b) Output (INR kgm^2)

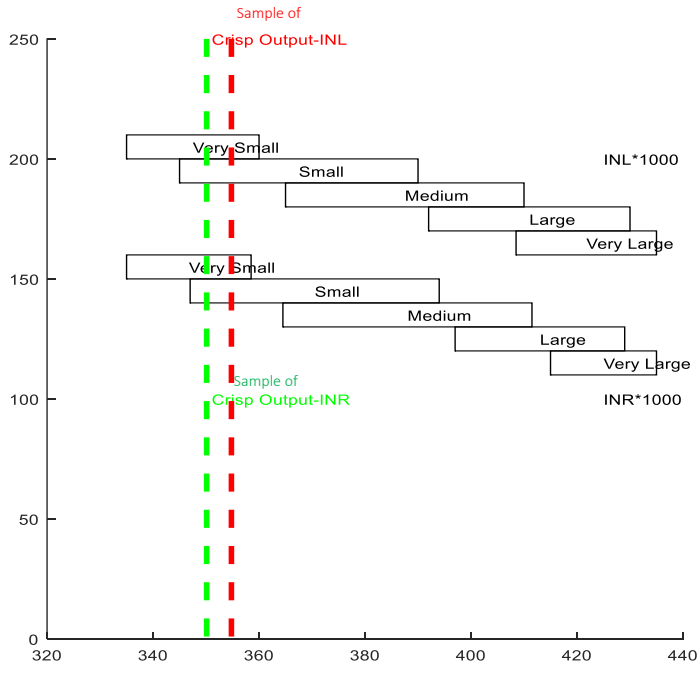


(c) Output (INL kgm^2)

Fig. 12 Membership functions for inertia estimation using laser simulator logic

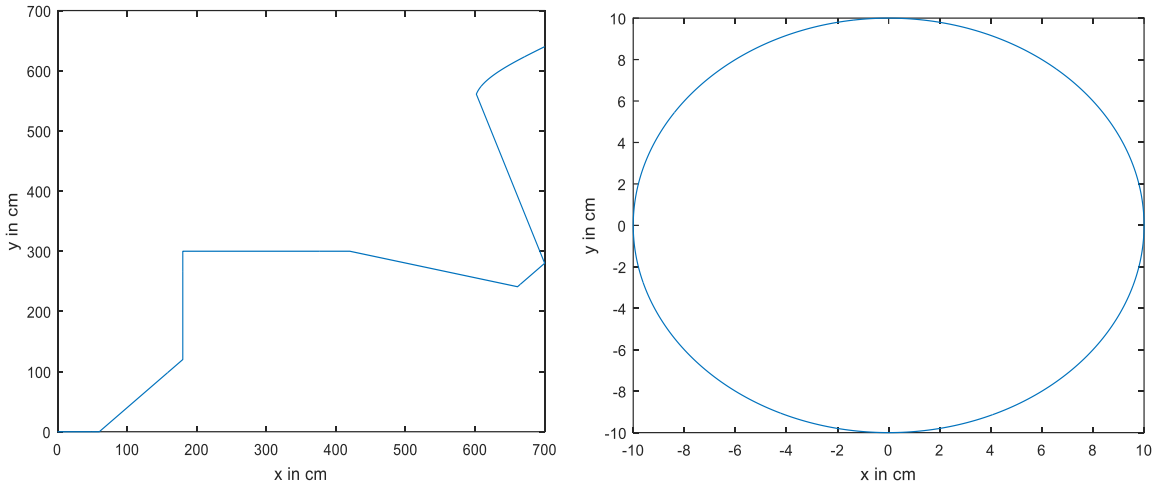


(a) input (φ^0)



(b) Output (INR and INL kgm^2)

Fig. 12 Membership functions for inertia estimation using laser simulator logic



(a) (b)

Fig. 13 Pre-planned path for testing the proposed control system in simulation (a) Zigzag path (b) circular path

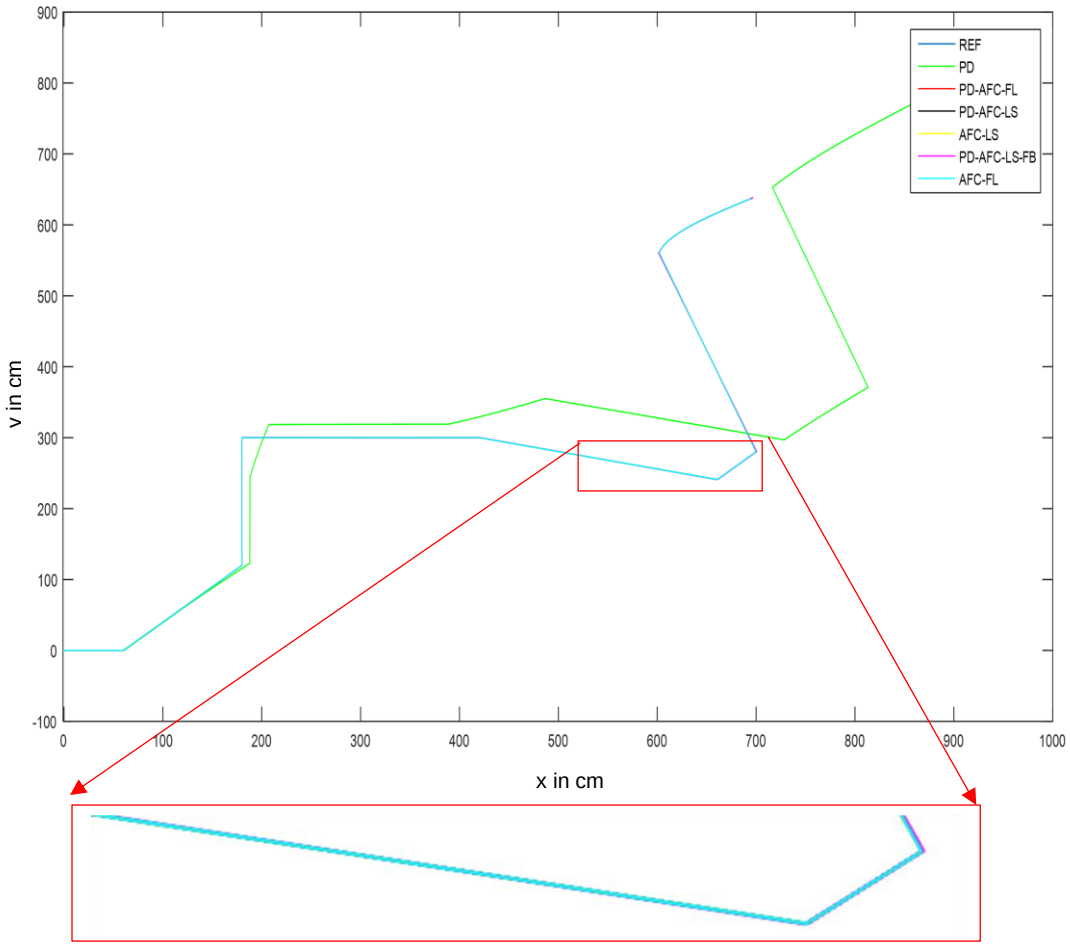
Fig. 14 shows the implementation of the controllers on the zigzag path without disturbance. All the controllers have tracked well the zigzag path as shown in Fig. 14(a), except for PD controller. It can be noticed that there are ~~an~~ accumulative errors raised from the beginning to the end of the path. However, other controllers are almost similar in relation to the reference trajectory. Thus they overlap each other along the path. The tracking errors in x and y direction for PD are at the level of 10^2 with maximum errors of 250mm as shown in Fig. 14 (b) and (c). On the other hand~~s~~, the difference between the actual and reference paths by AFC-FL, AFC-LS, PD-AFC-LS, PD-AFC-FL and PD-AFC-LS-QC is too small with a tracking error less than 0.30 mm in both x and y directions as shown in Fig. 14 (b), (c). Fig. 14 (d) and (e) show that the AFC-LS and AFC-FL have higher tracking errors in both x and y in comparison with PD-AFC-LS, PD-AFC-FL and PD-AFC-LS-QC controller with maximum tracking errors in the corners equal to 0.28mm. The PD-AFC-LS, PD-AFC-FL and PD-AFC-LS-QC have low tracking errors in both x and y directions with a maximum error that is equal to 2×10^{-3} mm as shown in Fig. 14 (f) and (g). The PD-AFC-LS and PD-AFC-LS-QC illustrate a better performance with tracking errors of less than 1×10^{-5} mm as shown in Fig. 14 (h) and (i). PD-AFC-LS-QC possesses the best performance of the control system with the capability to eliminate the noise that still occurs in PD-AFC-LS as shown in Fig. 14 (j) and (k).

Fig. 15 shows the implementation of the controllers on a circular path without disturbance. Similar to the zigzag path analysis, all controllers except for the PD controller can track the circular path with small errors. The tracking errors in x and y direction for PD have a maximum error of 5mm as shown in Fig. 15 (b,c). On the other hand~~s~~, the difference between the actual and reference paths by AFC-FL, AFC-LS, PD-AFC-LS, PD-AFC-FL and PD-AFC-LS-QC is too small with a tracking error of less than 0.25 mm in both x and y directions as shown in Fig. 15 (b), (c). Fig. 15 (d) and (e) show that the AFC-LS and AFC-FL have higher tracking errors in both x and y in comparison with PD-AFC-LS, PD-AFC-FL and PD-AFC-LS-QC controller with maximum tracking errors that are equal to 10^{-3}mm. The PD-AFC-LS, PD-AFC-FL and PD-AFC-LS-QC have low tracking errors in both x and y with a maximum error that is equal to 5×10^{-5} mm as shown in Fig. 15 (f)and (g). The PD-AFC-LS and PD-AFC-LS-QC show a better performance with tracking errors of less than 1×10^{-6} mm as shown in Fig. 15 (h) and (i). PD-AFC-LS-QC has the best performance of the control system with the capability to eliminate the noise that still exists in PD-AFC-LS as shown in Fig. 15 (j) and (k). By looking to zooming areas in Fig. 14 a and Fig. 15 a, one can see that there are slight differences between the AFC family controllers in the thickness of trajectories' lines which means there are some controllers that are oscillating along their trajectory such as AFC-FL, AFC-LS and PD-AFC-FL, and some others are robust enough in their path such PD-AFC-LS and PD-AFC-LS-QC.

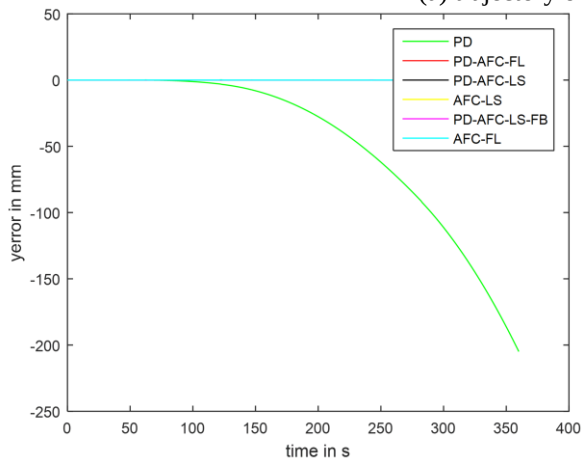
7.2 Simulation Results of Controllers in the Presence of Disturbances

A variable harmonic disturbance has been applied on the zigzag and circular paths.

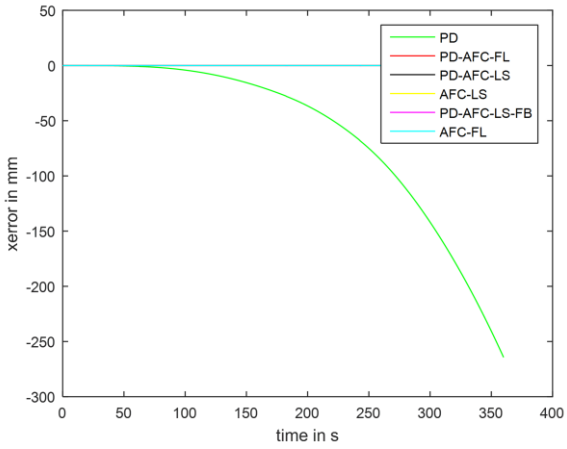
Fig. 16 shows the actual paths of all the controllers when they are tracking the zigzag path as shown in Fig. 13 with a high variable harmonic disturbance described by the equation: $\tau_d = \begin{bmatrix} 10^4\cos(t) \\ 10^4\cos\left(t+\frac{\pi}{2}\right) \end{bmatrix}$. PD controller presents an actual trajectory that is different from the reference by 10^6 times, so we didn't draw it in Fig. 16 as its trajectory is too far from the reference and other controllers trajectories, however other controllers have the trajectory that is almost similar to reference path. Thus they overlap ~~each other's~~ as shown in Fig. 16 (a). The tracking errors of the PD controller are in the range of 10^6 mm which can be considered as large tracking errors that can't be compensated by any kind~~s~~ of controller as shown in Fig. 16 (b) and (c). However, other controllers have small tracking errors in x and y directions with maximum errors of 0.32 mm as shown in Fig. 16 (d), (e) which are quite similar to the previous case when no disturbance has been applied.



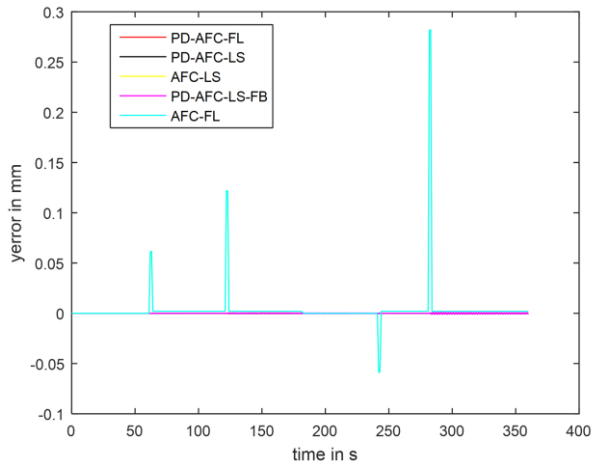
(a) trajectory of all controllers

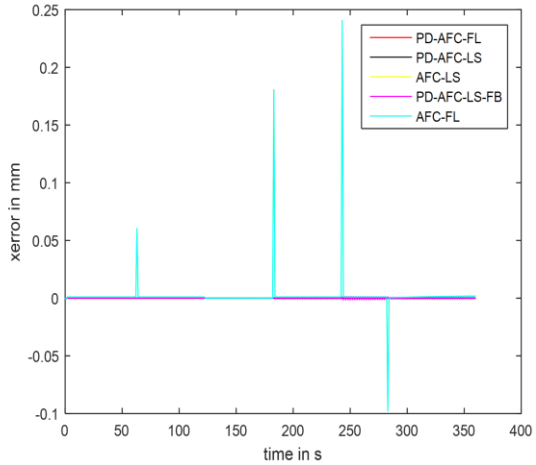


(b) *y* track errors for all controllers

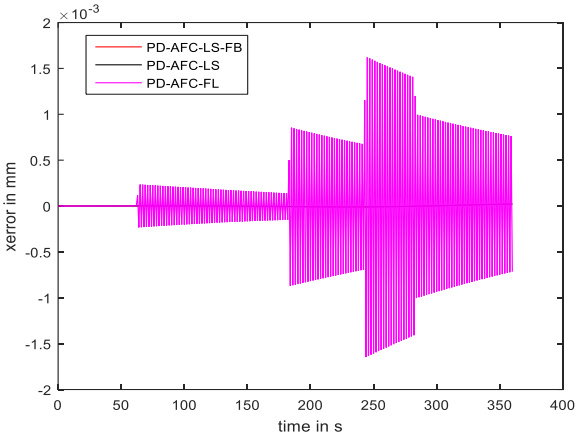


(c) *y* track errors for all controllers

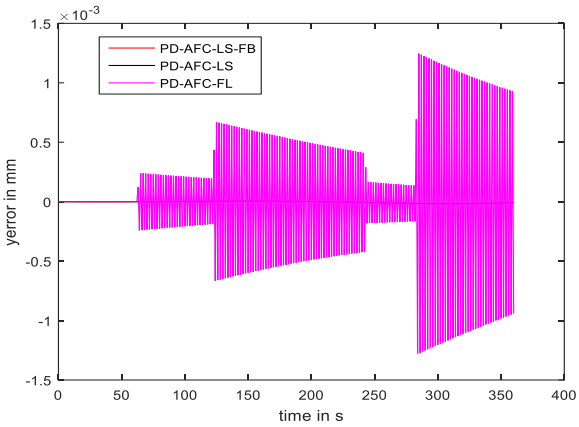




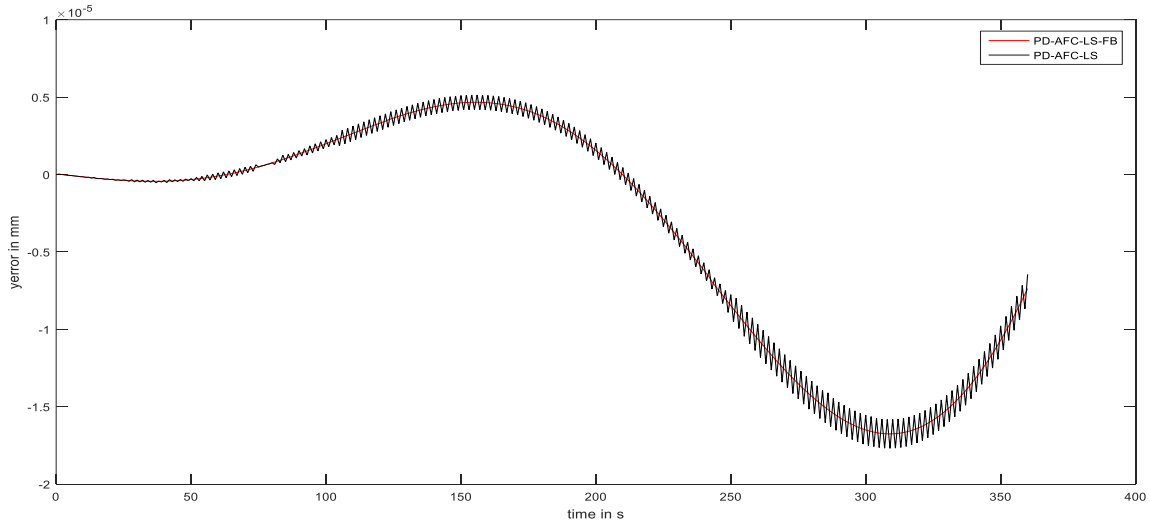
(d) *y* track errors of all controllers except PD (e) *x* track errors of all controllers except PD



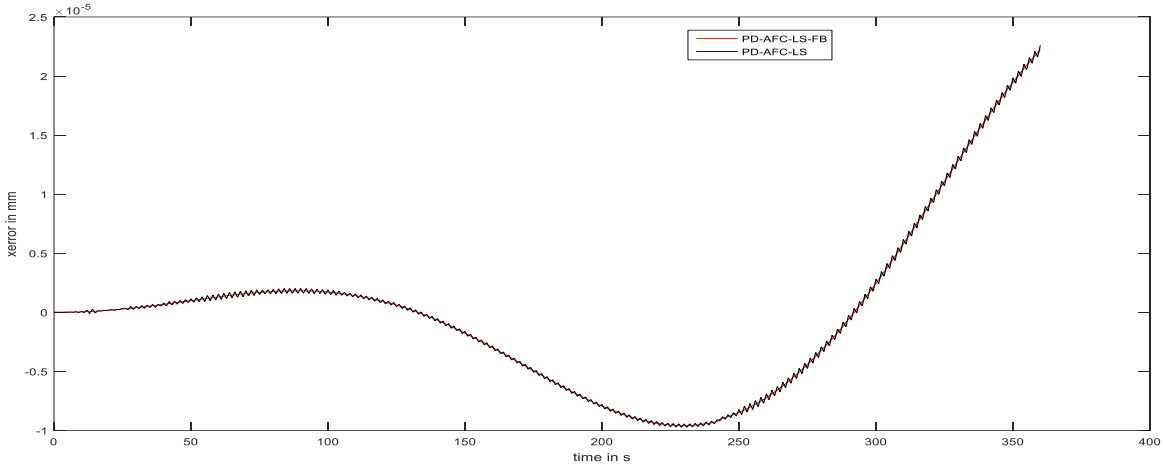
(f) *x* track errors for PD-AFC types



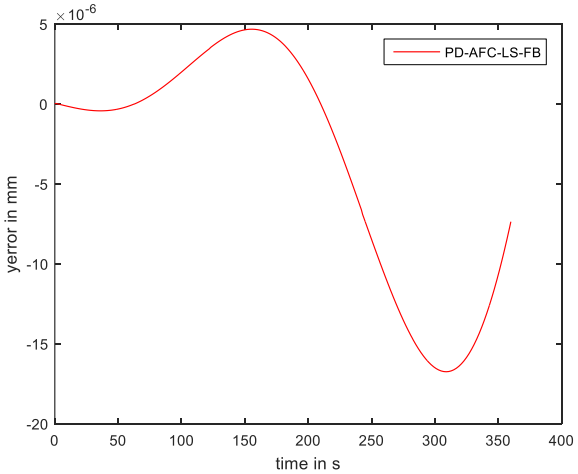
(g) *y* track errors for PD-AFC types



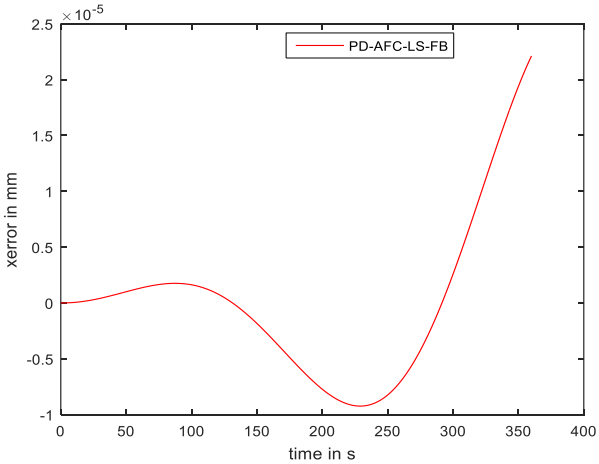
(h) y track error for PD-AFC (LS and LS-QC)



(i) x track error for PD-AFC (LS and LS-QC)

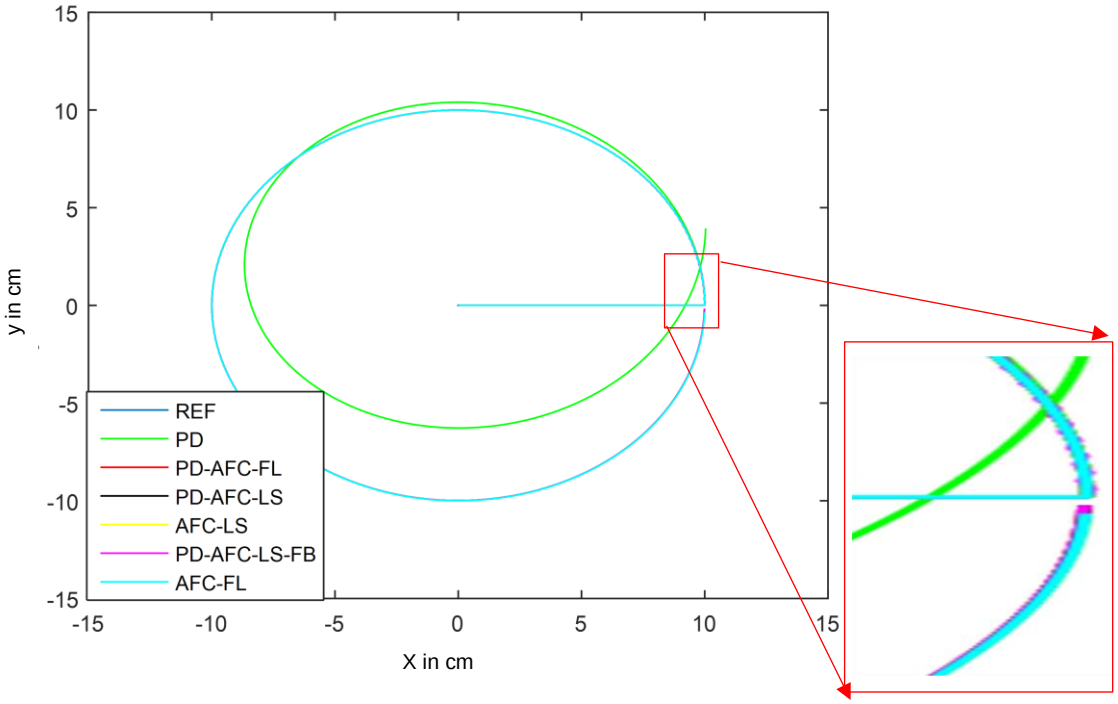


(j) y track error for PD-AFC-LS-QC

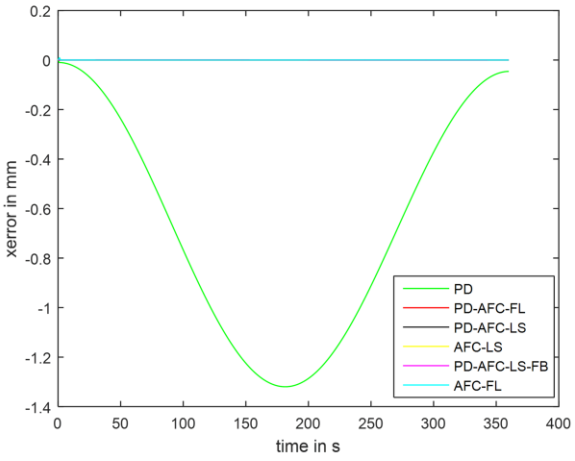


(k) x track error for PD-AFC -LS-QC

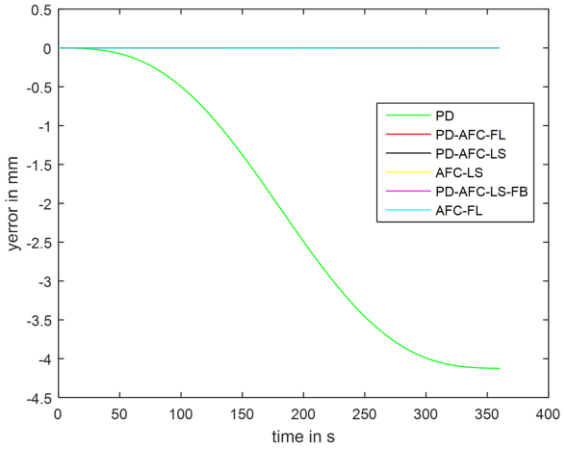
Fig. 14 Results for all controller schemes in zigzag path without disturbance



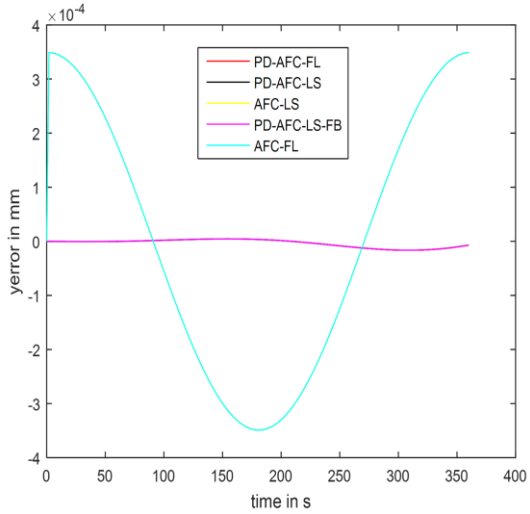
(a) the trajectory of all controllers



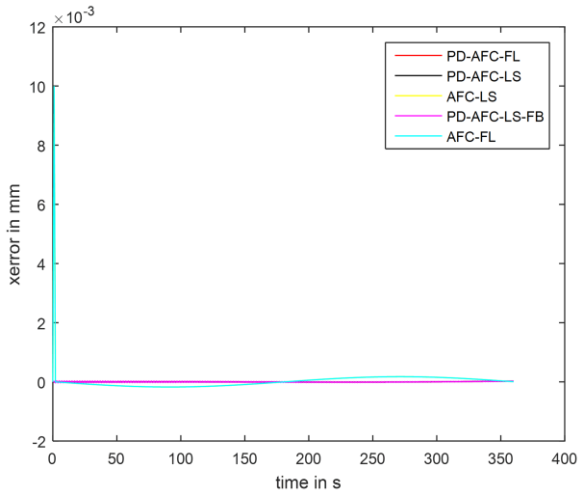
(b) *y* track errors for all controllers



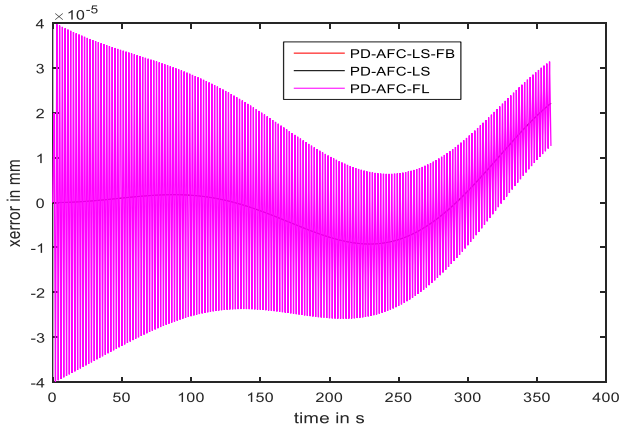
(c) *y* track errors for all controllers



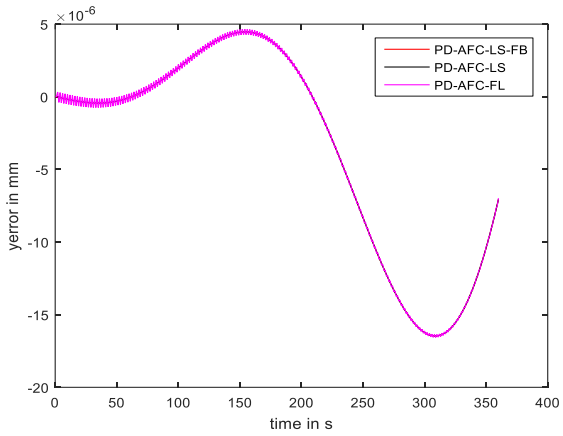
(d) *y* track errors of all controllers except PD



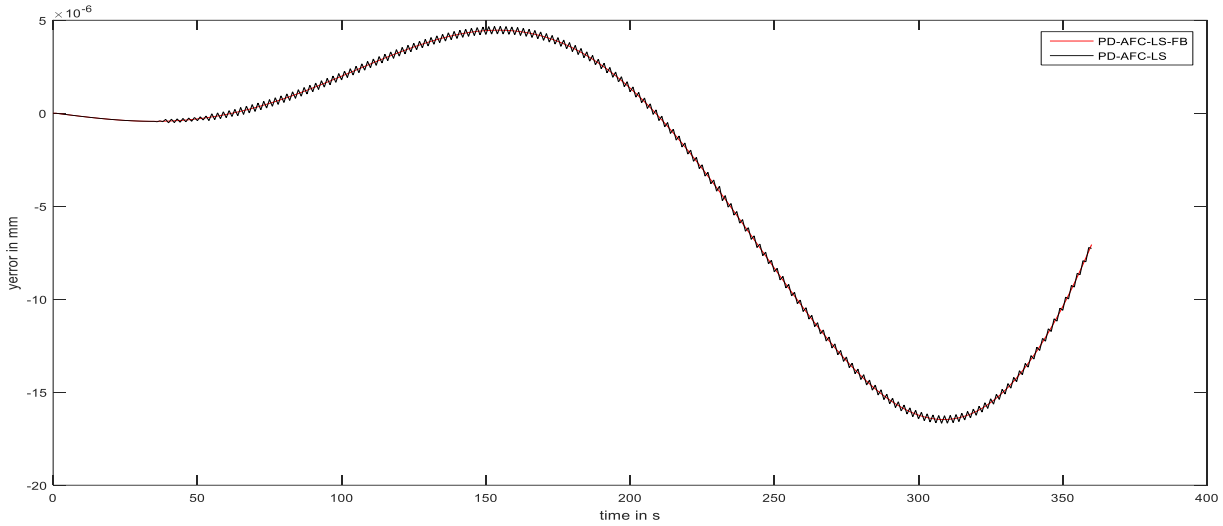
(e) *x* track errors of all controllers except PD



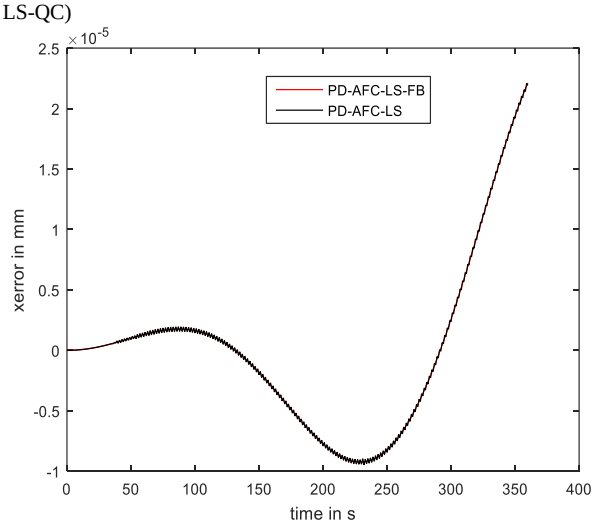
(f) *x* track errors for PD-AFC types



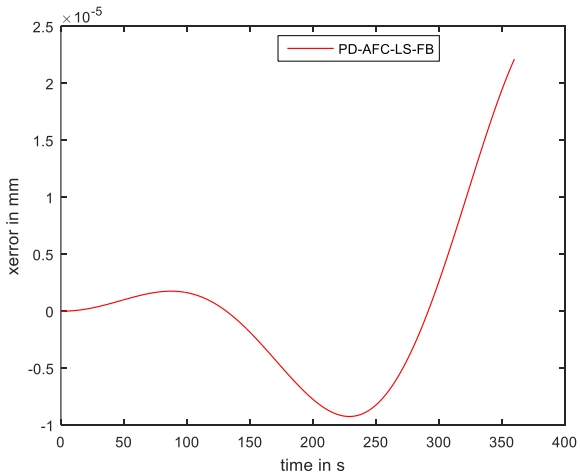
(g) *y* track errors for PD-AFC types



(h) y track error for PD-AFC (LS and LS-QC)



(i) x track error for PD-AFC (LS and LS-QC)



(j) y track error for PD-AFC-LS-QC (k) x track error for PD-AFC -LS-QC

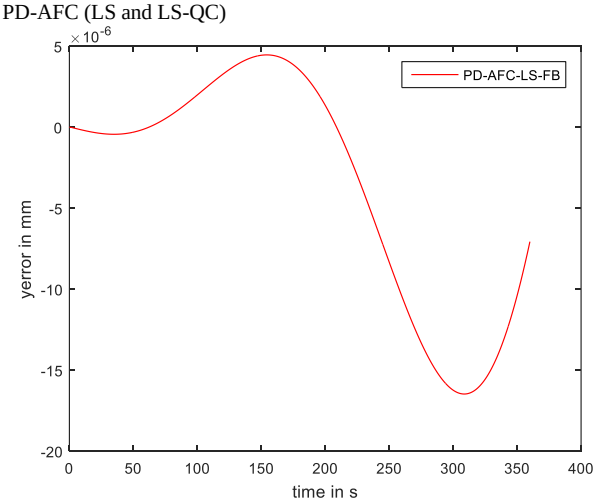


Fig. 15 Results of all controllers' schemes in circular path without disturbance

There is a sensible difference between the AFC controllers (with LS and FL) and PD-AFC controller (with LS and FL) as shown in Fig. 16 (d) and (e) where the tracking errors in PD-AFC are almost zero along with the motion. However, AFC has errors in the region of 0.25mm. PD-AFC-FL, PD-AFC-LS and PD-AFC-LS-QC presents a good response against the applied disturbance with maximum errors of 10^{-3} mm as shown in Fig. 16 (f) and (g). The proposed PD-AFC-LS-QC controller has possessed the best performance against the disturbances among all controllers as shown in Fig. 16 (h), (i) with a maximum error of 10^{-6}mm.

Fig. 17 shows actual paths of all the controllers when they are tracking the circular path with quite high variable disturbance described by the following equation:

$$\tau_d = \begin{bmatrix} 10^4 \cos(t) \\ 10^4 \cos\left(t + \frac{\pi}{2}\right) \end{bmatrix}.$$

Similar to the case mentioned above, the PD controller leads to an actual trajectory which causes the actual path to be totally different from the reference, thus it is not drawn in Fig. 17 as there is no similarity at all between PD's and the original trajectories; however other controllers have the trajectory which is exactly similar to the reference path as shown in Fig. 17 (a). The tracking errors of the PD controller are in the range of 10^6 mm which can't be compensated in any way as shown in Fig. 17 (b) and (c). However, the others controllers have small tracking errors in x and y directions with a maximum error of 12×10^{-3}mm as shown in Fig. 17 (d), (e) which are quite similar to the previous case when no disturbance has been applied. There is a sensible difference between the AFC controllers (with LS and FL) and PD-AFC controller (with LS and FL) as shown in Fig. 17 (d) and (fe where the tracking errors in PD-AFC are almost zero along with the motion. However, AFC has errors in the region of 10^{-3}mm. PD-AFC-FL, PD-AFC-LS and PD-AFC-LS-QC have a good response against the disturbance with maximum errors of 10^{-5} mm as shown in Fig. 17 (f) and (g).

The proposed PD-AFC- LS-QC controller presents the best performance against the disturbance among all controllers as shown in Fig. 17 (h) and (i) with a maximum error of 10^{-6}mm. same analysis goes to the zooming areas in Figs. 16 a and 17 a, where there are slight differences between the AFC family controllers in the thickness of trajectories' lines which mean there are some controllers that are oscillating along their trajectory such as AFC-FL, AFC-LS and PD-AFC-FL, however some others are robust enough in their path such PD-AFC-LS and PD-AFC-LS-QC.

x track errors for PD-AFC types (g) *y* track errors for PD-AFC types

The mean tracking errors for zigzag and circular path without/with disturbances are illustrated in Tables 6 and 7 respectively. From these tables, we can see that only PD controller is affected when applying disturbances, however other controllers are able to robustly eliminate the effect of disturbances, especially PD-AFC-LS-QC.

8. Real-Time Mobile Robot Control

The PD-AFC-LS-QC has been tested in a real-world environment in real-time as follows:

8.1 Experiment setup

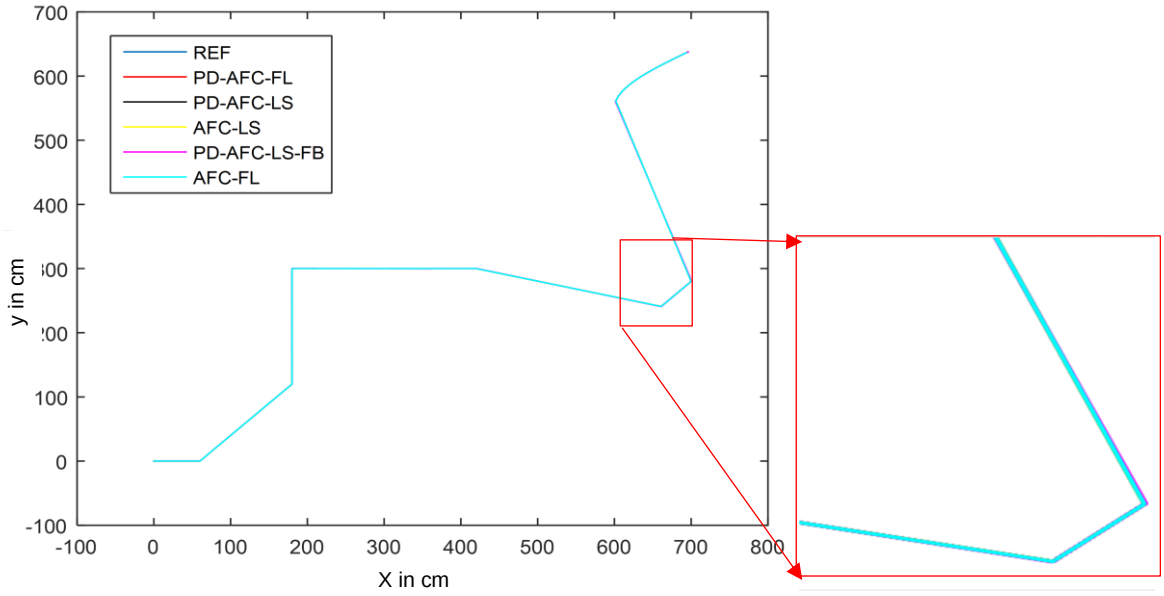
A medium-size wheeled mobile robot (WMR) platform as shown in Fig. 18 has been built in the lab to perform the control system in real-time. It is equipped with three units to apply the proposed control system on the robot, namely, measurement, processing and driving units. The measurement unit uses two sensors, namely, encoders and the current sensor for measuring the actual acceleration and torque of AFC. The processing unit is used to prepare the signal and data in the useful form before it is utilized in the AFC calculations. The driving unit acts as the controller through PWM signals and motor drivers on the DC motors.

Table 6: Mean tracking errors of zigzag path (in mm)

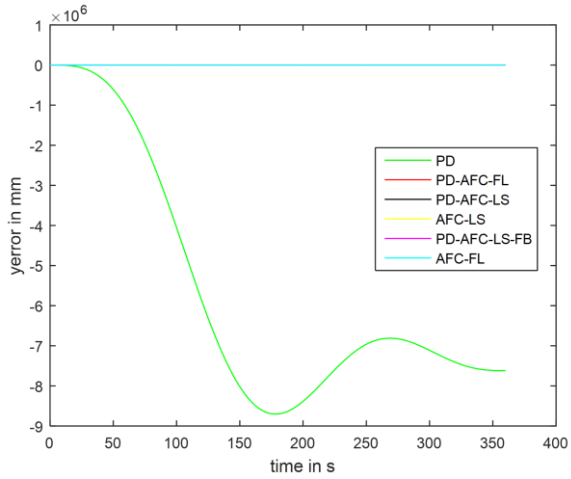
	Without disturbance						With disturbance					
Mean tracking errors (mm)	PD	AFC-FL	AFC-LS	PD-AFC-FL	PD-AFC-LS	PD-AFC-LS-QC	PD	AFC-FL	AFC-LS	PD-AFC-FL	PD-AFC-LS	PD-AFC-LS-QC
X track errors	-125	0.7	0.06	-0.05×10^{-3}	0.75×10^{-5}	0.65×10^{-5}	-2.8×10^{6}	0.7	0.06	-0.05×10^{-3}	0.75×10^{-5}	0.65×10^{-5}
Y track errors	-100	0.115	0.075	-0.05×10^{-3}	-0.55×10^{-5}	-0.5×10^{-5}	-4.25×10^{6}	0.11	0.075	-0.05×10^{-3}	-0.55×10^{-5}	-0.5×10^{-5}

Table 7: Mean tracking error of circular path (in mm)

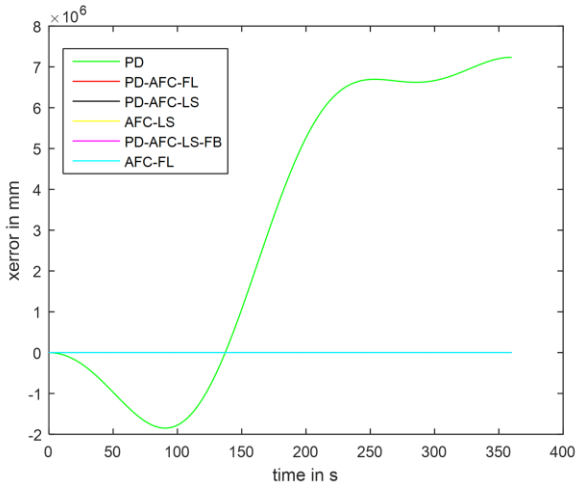
	Without disturbance						With disturbance					
Mean tracking errors (mm)	PD	AFC-FL	AFC-LS	PD-AFC-FL	PD-AFC-LS	PD-AFC-LS-QC	PD	AFC-FL	AFC-LS	PD-AFC-FL	PD-AFC-LS	PD-AFC-LS-QC
X track errors	-0.65	5×10^{-3}	4×10^{-3}	0.1×10^{-5}	0.75×10^{-5}	0.7×10^{-5}	-2.8×10^{6}	5×10^{-3}	4×10^{-3}	0.1×10^{-5}	0.75×10^{-5}	0.7×10^{-5}
Y track errors	-2	0.01×10^{-3}	0.05×10^{-4}	-0.65×10^{-5}	-0.5×10^{-5}	-0.6×10^{-5}	-4.25×10^{6}	0.01×10^{-3}	0.05×10^{-4}	-0.65×10^{-5}	-0.5×10^{-5}	-0.6×10^{-5}



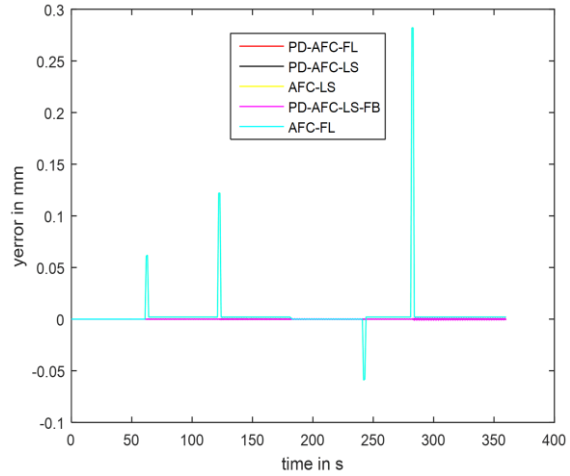
(a) trajectory of all controllers except PD controller



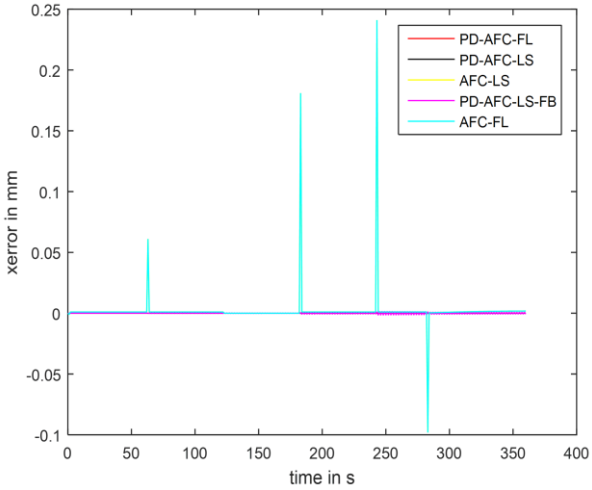
(b) *y* track errors for all controllers



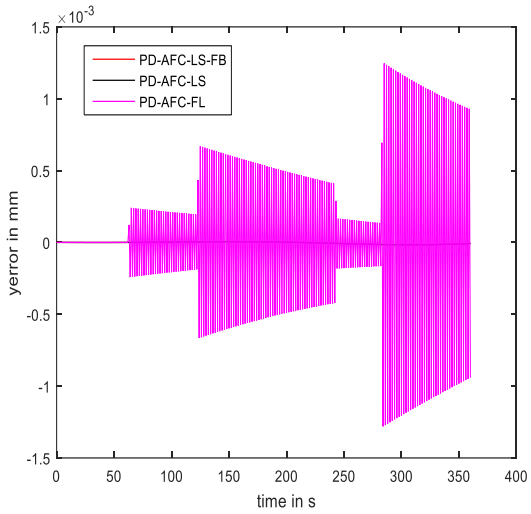
(c) *x* track errors for all controllers

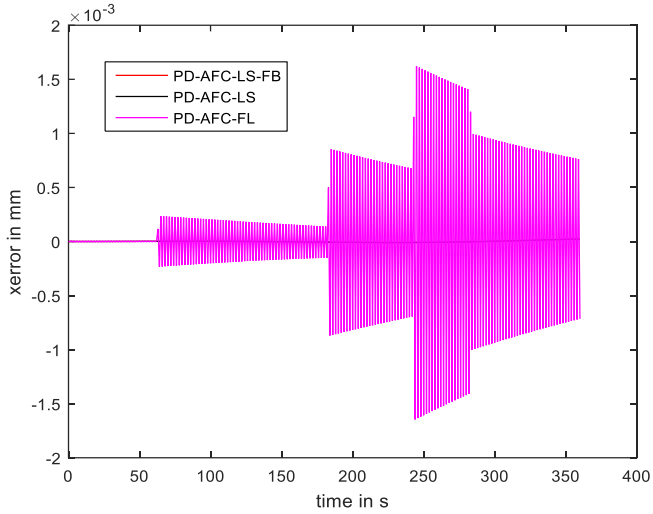


(d) *y* track errors of all controllers except PD

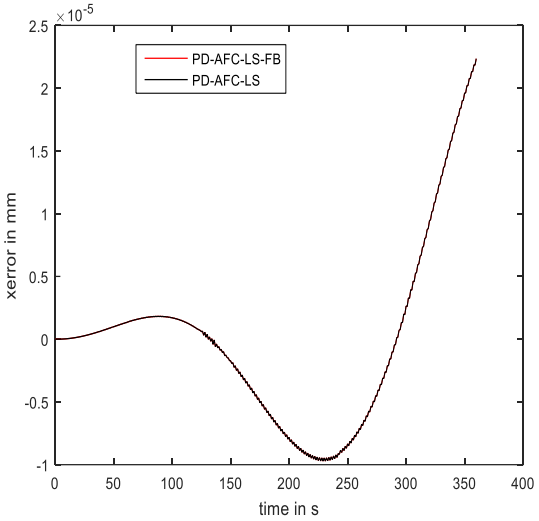


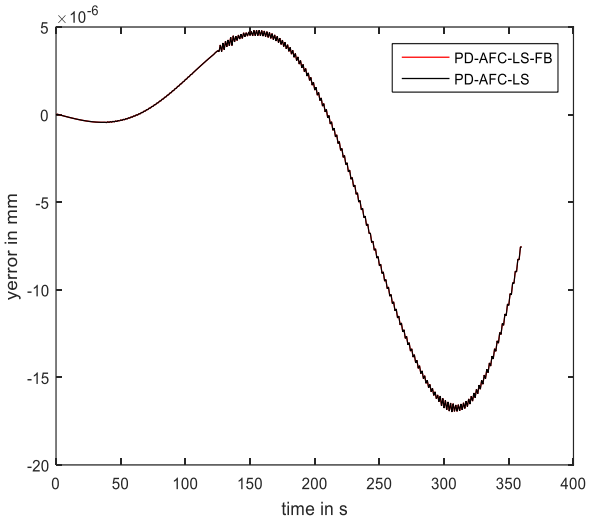
(e) *x* track errors of all controllers except PD





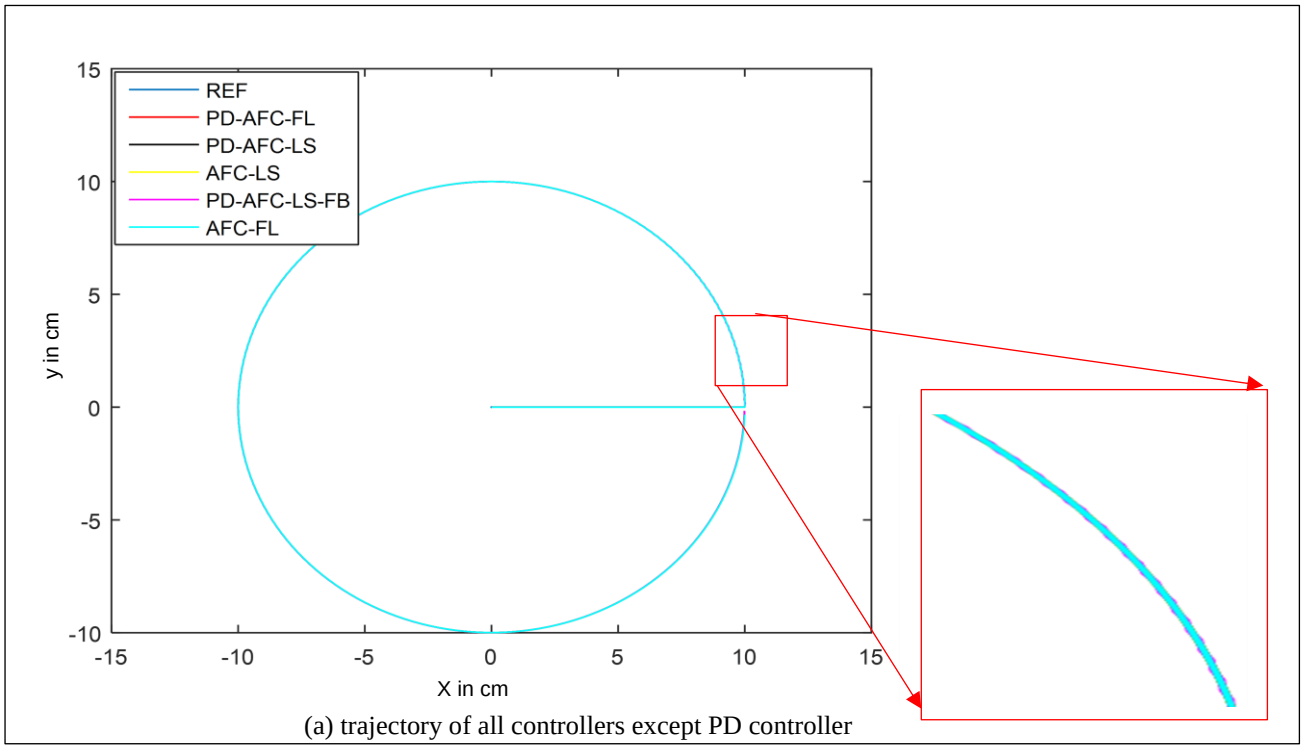
(f) *x* track errors for PD-AFC types (g) *y* track errors for PD-AFC types



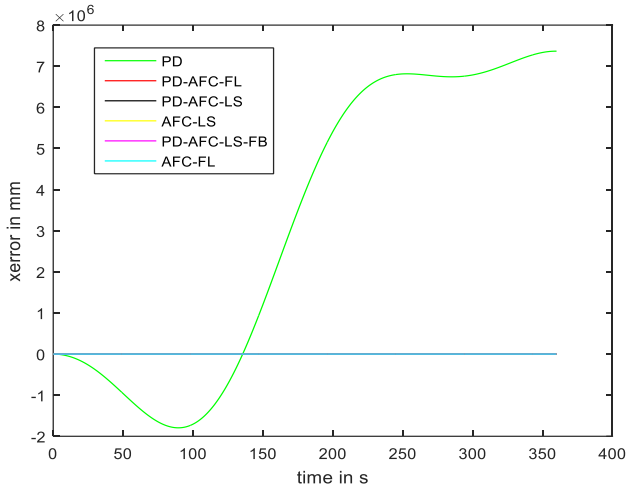


(h) y track error for PD-AFC (LS and LS-QC) (i) x track error for PD-AFC (LS and LS-QC)

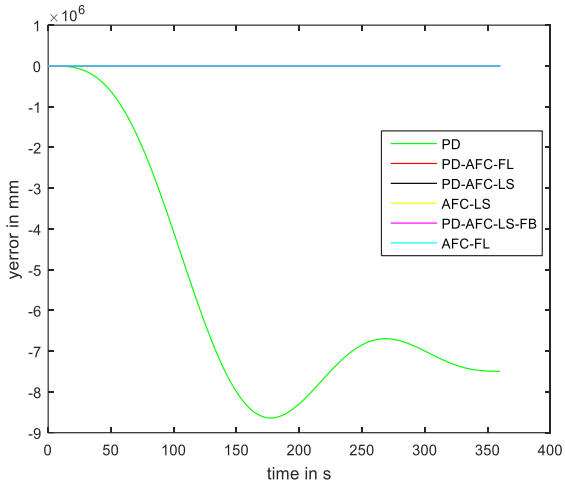
Fig. 16 Results for all controllers schemes in zigzag with disturbance $\tau_d = \begin{bmatrix} 10^4\cos(t) \\ 10^4\cos\left(t+\frac{\pi}{2}\right) \end{bmatrix}$ Nm



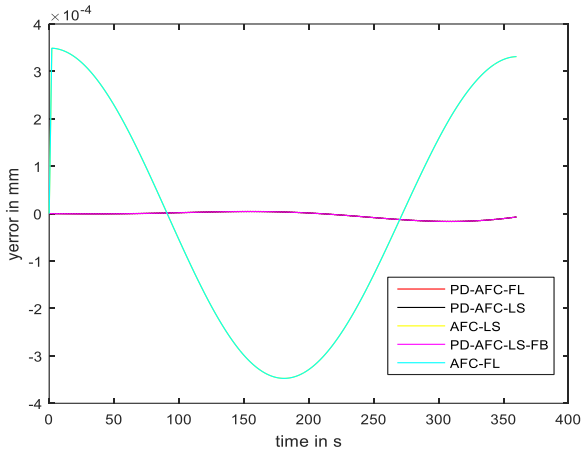
(a) trajectory of all controllers except PD controller



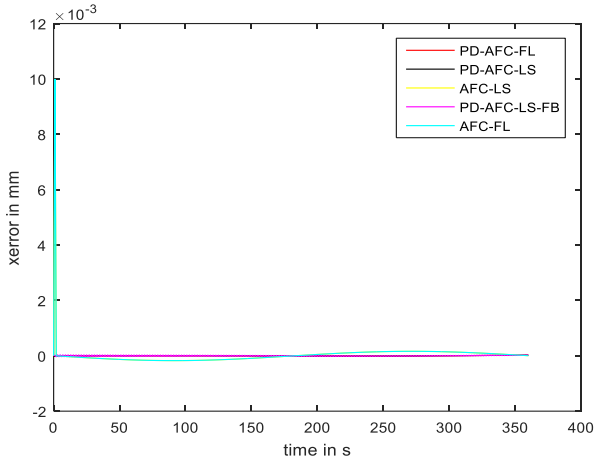
(b) *y* track errors for all controllers



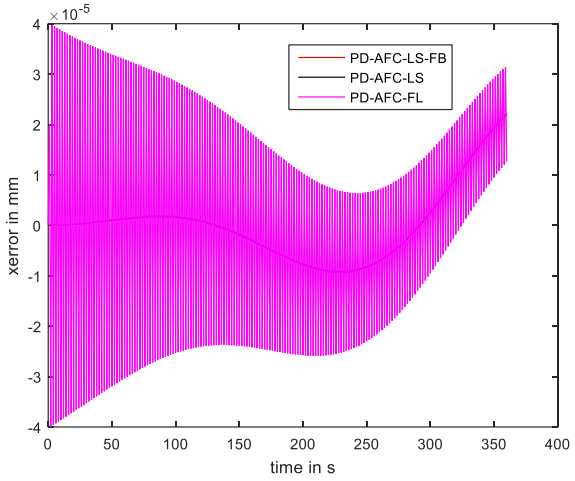
(c) *y* track errors for all controllers



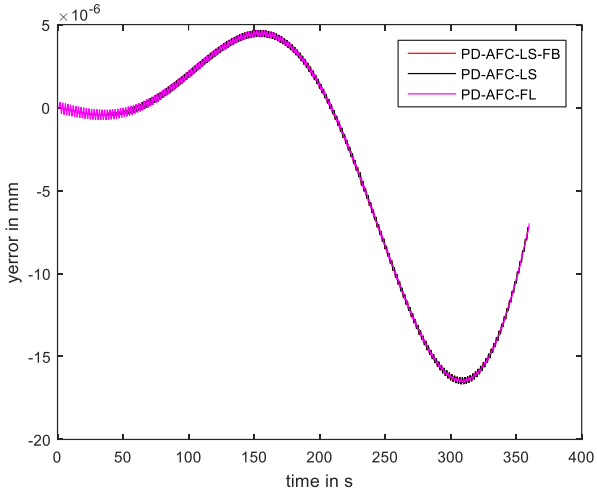
(d) *y* track errors for of all controllers except PD



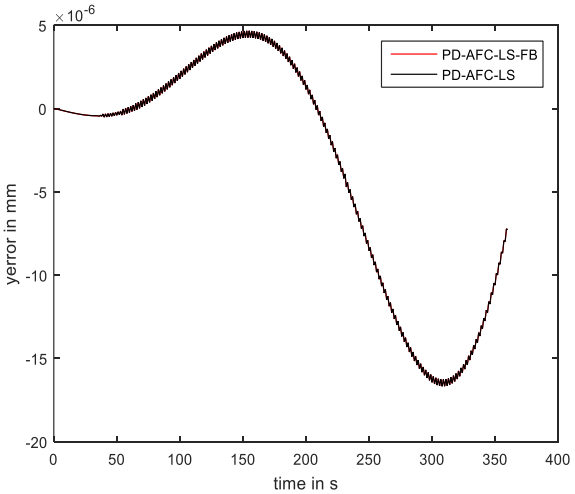
(e) *x* track errors for of all controllers except PD



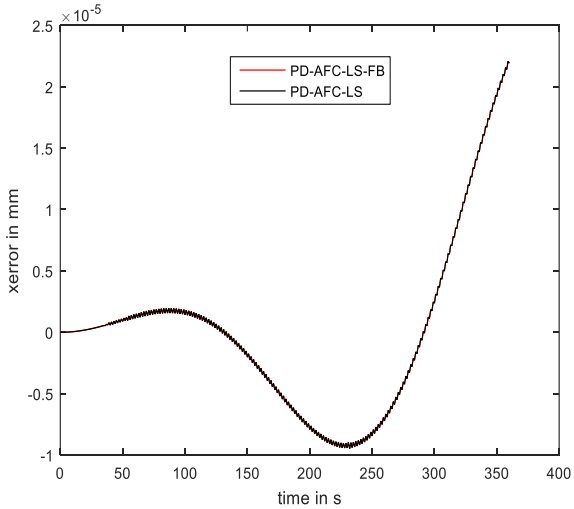
(f) *x* track errors for PD-AFC types



(g) *y* track errors for PD-AFC types

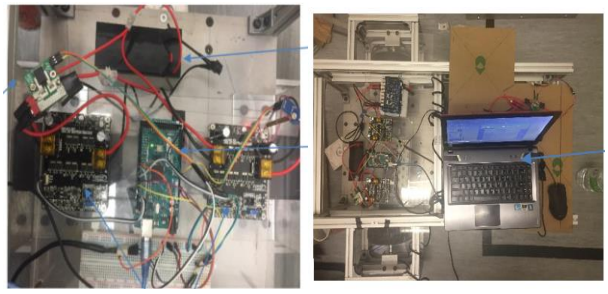


(h) y track error for PD-AFC (LS and LS-QC)

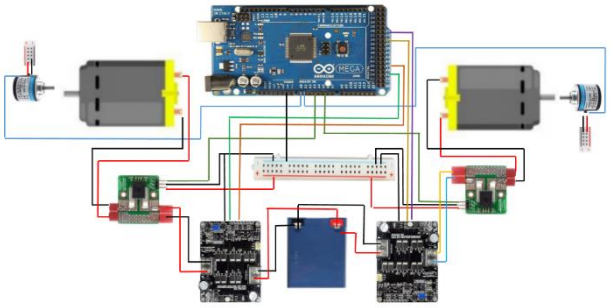


(i) x track error for PD-AFC (LS and LS-QC)

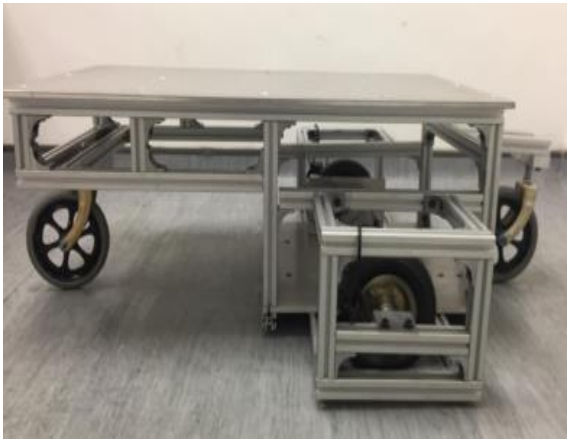
Fig. 17: Results of all controllers' schemes in a circular path with disturbance



(a) sensor, cards, motor driver and On board PC connection



(b) Connection between processing units (red color for current and other colors for data)



(c) Platform chassis components

Fig. 18 Embedded system for the robot platform

The proposed PD-AFC-LS-QC controller has been used to track the robot in a circular path with a diameter of 1m. This controller has been selected since it presents the best performance in the simulation study among all the other controllers. Similar to the simulation study, this controller has three loops, namely, external, internal and quick compensation loops.

In this control system, the current sensors are used to measure the applied torque on the motors due to the linear relation between the torque and current in brushed DC-motors DKM $\tau = k_t I$. K_t is the torque constant which can be determined easily in DKM-DC motors from the characteristics of torque-current as shown in Fig. 19.

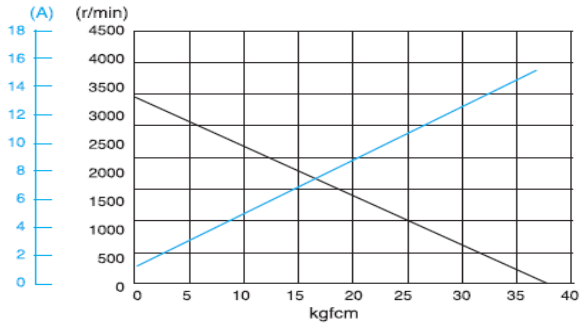


Fig. 19 Torque-current characteristic of *DKM*-motors (120 W) (White Drive Products, 2005)

The rotary encoders have been used to claculate the angular acceleration of the wheels. Based on the measurement of the encoder, one can calculate the angular position of the wheels, in rpm by Eq. 51

$$n = \frac{P_{cur}}{T\ P_{rf}} \tag{51}$$

P_{cur} is the number of encoder pulses for the whole robot wheel movement from the initial to current positions, P_{rf} is the full rotation's pulses number, T is the sampling time of acquiring data.

The acceleration is then computed by applying a second derivation on every two successive measurements of the encoder (Ali, 2019b) as in Eq. 52:

$$a = \frac{\partial_2 (x_2 - x_1)}{\partial T^2} \tag{52}$$

In the experimental work, the Laser Simulator Logic has been used for estimating the inertia matrix, since it has been proven in simulation in Section 7, that Laser Simulator Logic can estimate effectively the inertia moment better than fuzzy logic. On the other hand, the disturbances are estimated using Eq. 46.

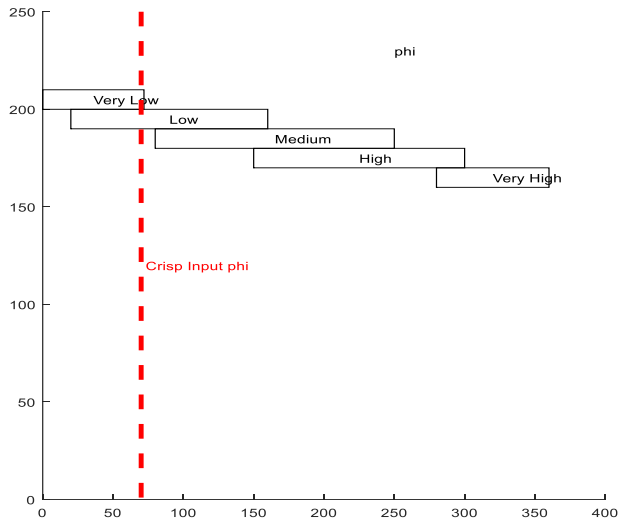
The WMR sensors measurements and the inertia moment calaculations are used to estimate the actual torque of the motor which is later comapred with the applied torque to find the disturbance torque as seen in Eq. 40. Similar to the simulation, the Laser Simulator linguistic variables and memebership functions, have a highly overlapped condition where there are some linguistic variables covering a large range in the universe of discourse. However, some others have a small range. The input LS set is: φ={Very-low, Low, Medium, High, Very-High} as shown in Fig. 20 (a). The output LS sets are: INR-INL={ Very-Small, Small, Medium, Large, Very-Large} as shown in Fig. 20 (b).

WMR Parameters:

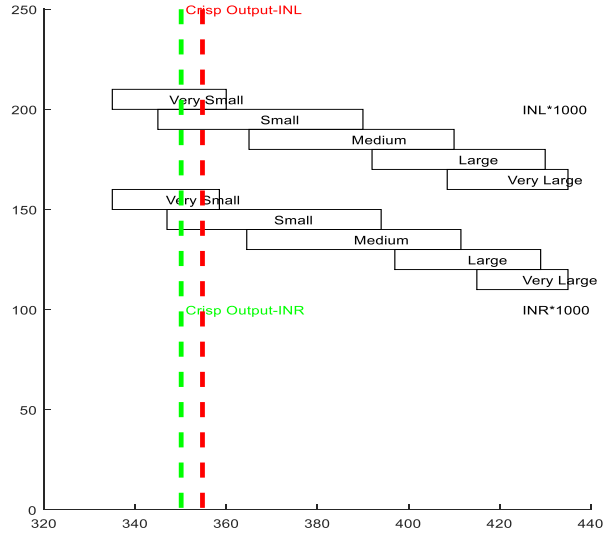
r = 105.4 mm, b = 500 mm, d = 16.5 mm, m = 36 kg, m_w= 1.5kg, I =10.0567 kg.m^2 , I_w= 0.0083 kgm^2, V = 0.15m/s.

Controller Parameters:

K_{px} = 2/s^2, K_{py} = 2/s^2, $K_{p\varphi}$= 2/s, K_{dx} = 1, K_{dy} = , $K_{d\varphi}$= 1 , K_n = 0.00108 Nm/rpm



(a) Input (φ)



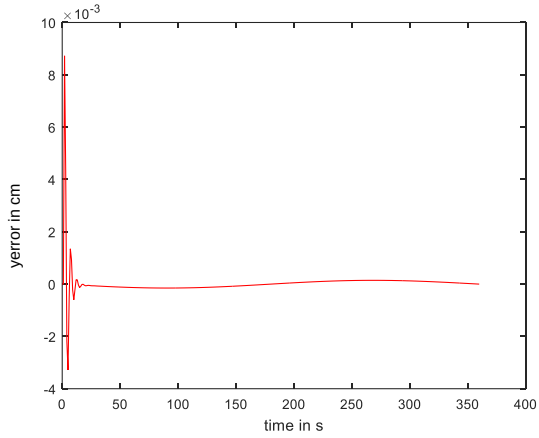
(b) Output INR and INL

Fig. 20 Membership functions of the LS: (a) input set (φ) (b) output set INR/INL

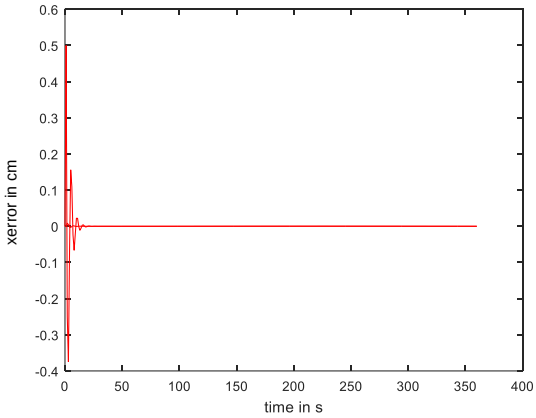
8.2 Experiments' Results and Discussion

In this work, it has been considered that the slippage of the lab ground is the disturbance of the PD-AFC-LS-QC control system. The robot has been forced to track a circular path with a diameter of 1m. Fig. 21 shows the capability of the robot to track the circular path. The tracking errors in both *x* and y directions have been shown in Fig. 22. In comparison with simulation, the tracking errors become more prominent, but they are still in an acceptable range. PD-AFC-LS-QC has tracking errors in the y direction in the region of 10^{-3} mm, which is bigger than the simulation value by around two times with a maximum value equal to 9×10^{-3} mm that occurs at the beginning of the movement. On the other hand, the tracking errors in the x direction are bigger than tracking errors of y. It is in the region of 10^{-1} mm, which is different compared with the simulation where the errors were in the region of 10^{-5}mm. It is noticeable that the errors of x become zero after the settling time. However, in y-direction, they fluctuate slightly before they become stable and reach zero. In general, the results show that the proposed PD-AFC-LS-QC controller is capable of maintaining the robot on its path and reject the noises.

Fig. 21: A sequence of WMR movements on a circular path



(a) *y* track errors



(b) x track errors

Fig. 22 Real-time WMR tracking errors when moving in a circular path

The robot has been tested in the other two environments, namely indoor (as shown in Fig. 23) and outdoor environments, (as shown in Fig. 25). In the indoor experiments, the robot has an obstacle in-front of its path which needs to be avoided. This forms a significant disturbance to the control system which aims to maintain the robot in the middle path between the curbs, in a long movement. Fig. 23 shows three types of paths:

- Red continuous line: the optimum path for navigation (without errors)

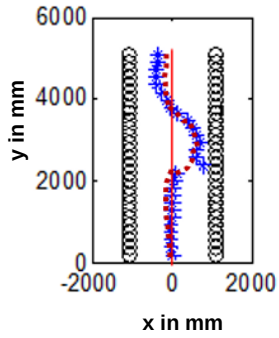
- Dotted maroon points: path planning of the robot
- Star blue points: actual path control.

It can be seen from Fig. 23 that the robot can avoid the obstacle and track the path with small errors. It has been noticed that the robot undergoes a jump when detecting the obstacle. This is due to that the algorithm is busy with obstacle detection and doesn't calculate the path at that moment. The path started to be determined after the robot confirms to pass the obstacle.

Fig. 24 shows the tracking errors in both x and y directions for the indoor application which is at the level of 1.5 mm and 1 mm, respectively.

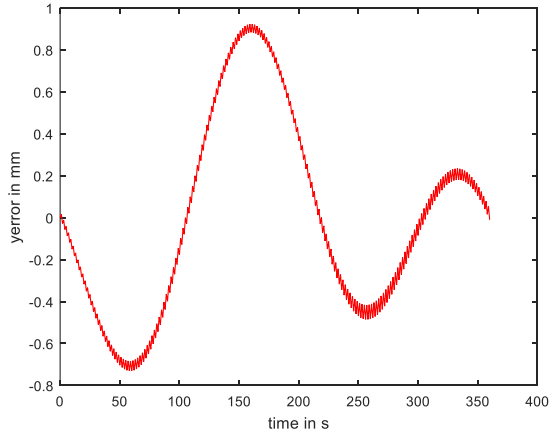
(a)

(b)

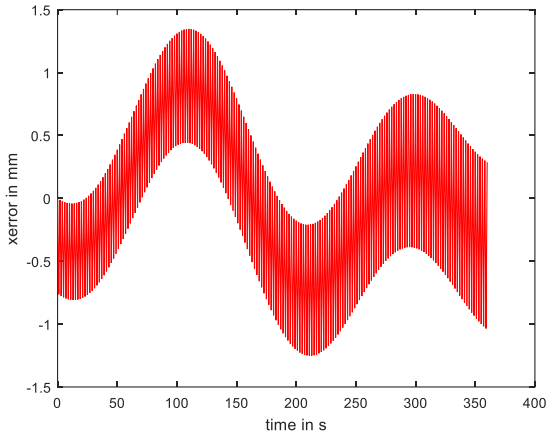


(c)

Fig. 23: Robot control in indoor path navigation with obstacles: (a) robot at the starting position. (b) robot avoiding obstacle (c) robot control path (red line for the optimum path, red dotted points for the path planning, and star blue for the actual path after applying control system)



(a) *y* track errors

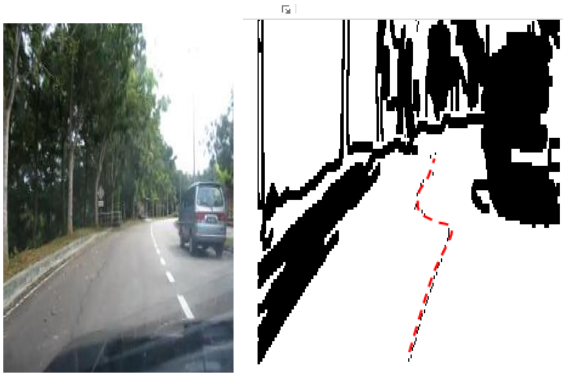


(b) *x* track errors

Fig. 24 Real-time WMR tracking errors when in indoor path control with obstacle

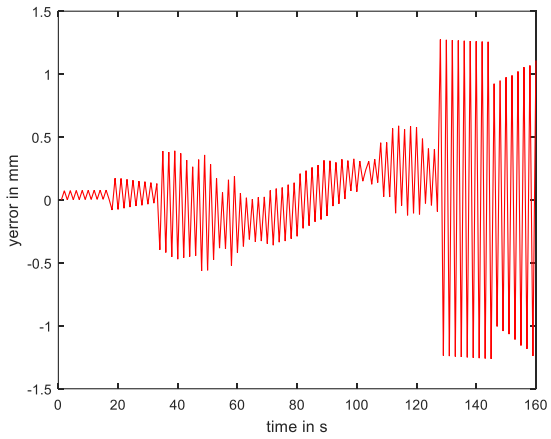
In the outdoor experiments, a unicycle-like robot system has an obstacle on the right side of its path which requires the robotic system to move to the left. Otherwise, it can crash on the side car. The path is then created and the control system must be able to avoid the obstacle and reject its effect on the path which is to go in the middle of the road. Fig. 25 shows a path control in road environment with obstacle avoidance. Two paths are generated, namely the planner path (red dash) and control path (black dash) after applying control system. Control path in Fig. 25 shows that the robot can track the path with the presence of an obstacle to avoid the obstacle with small errors.

Fig. 26 shows the tracking errors in both x and y for outdoor applications which are at the level of 5mm and 1.5 mm, respectively.

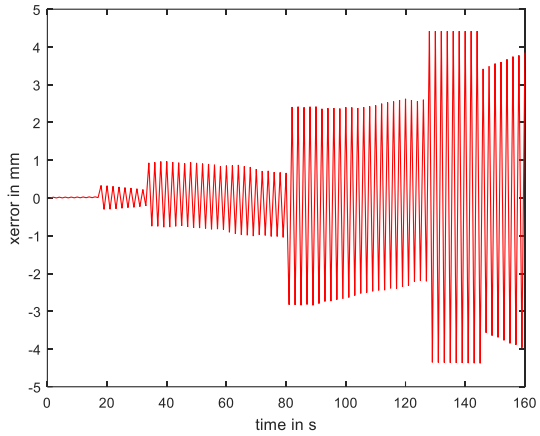


(a) (b)

Fig. 25: Unicycle-like robot control in outdoor path navigation with obstacles: (a) robot at the starting position. (b) robot control path with obstacle avoidance (red dots the path planning and black dash is the path after applying control system)



(a) *y* track errors



(b) *x* track errors

Fig. 26 Real-time WMR tracking errors when move-in outdoor path with an obstacle

A comparison between the mean tracking errors of the previous three experiments is illustrated in Table 8.

Table 8 : Mean tracking error (mm)

	Circular Path	**Indoor Path with Obstacle**	**Outdoor Path with Obstacle**
X track error (mm)	0.5	1.5	5
Y track error (mm)	8×10^{-3}	1	1.5

9. Conclusion

A novel algorithm for estimating the inertia matrix of AFC in a noisy environment and highly overlapped linguistic variable has been developed in this research. This algorithm is capable of accurately estimating the inertia matrix even though the linguistic variables are located in asymmetric distribution on the universe of discourse. A simulation study considers the use of Laser Simulator Logic to estimate the inertia moment in many controllers has been performed in Simulink and a comparison with fuzzy logic has been made.

Laser Simulator Logic shows a better performance in comparison with fuzzy logic in all the simulation study. A real-time experiment by WMR has been conducted to show the evidence of Laser Simulator in the real-time, which presents a good estimation of the inertia matrix in the presence of disturbances.

A medium-size wheeled mobile robot (WMR) platform has been built in the lab to perform the control system in real-time. It has been tested successfully with a circular path that has been developed in the lab to check the performance proposed algorithm in real-time.

A new robust control scheme, ~~so~~ called PD-AFC-LS-QC has been proposed and tested in the simulation and experimental works. In contrast with the traditional AFC controller that uses just two control loops, this controller uses three loops, namely, external, internal and quick compensation loops. In the external loop, a PD controller is used to control the kinematic parameters of the control system. Whereas, the internal loop is used to control the disturbances and dynamics of the robot. The quick compensation loop is used to quickly compensate for the difference between the actual and reference accelerations. The results of the simulation show that the proposed algorithm has the best performance among all the controllers either in zigzag or circular environments, mainly when the disturbances are applied. In order to confirm the results of the simulation for the proposed algorithm, a real-time experiments in three environments, namely the circular, indoor and outdoor paths have been conducted to show that the proposed controller scheme is robust enough in the real-time control and able to track the robot effectively on its reference path.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of following financial support for the research, authorship, and/or publication of this article: This research sponsored by Universiti Malaysia Pahang (UMP) and Ministry of High Education-Malaysia (MOHE) grants No: RDU180323, RDU1803138 and RDU190804.

References

Abdullah S, Mailah M and Hing T (2015) Feedforward Model-Based Active Force Control of Mobile Manipulator using MATLAB and MD Adams. *WSEAS Transactions on Systems* 12(6):*314-324.*

Abhishek V and Saha SK (2016) Dynamic Identification and Model based Control of an Omni-wheeled Mobile Robot. In: *4th International Conference on Robotics and Mechatronics*, Tehran, Iran, 26-28 Oct. 2016. PP. 595-600. USA: IEEE.

Ali MAH (2014) Autonomous Mobile Robot Navigation and Control in the Road Following and Roundabout Environments Incorporating Laser Range Finder and Vision System. PhD Thesis, UTM, Malaysia.

Ali MAH and Mailah M (2019a) A Simulation and Experimental Study on Wheeled Mobile Robot Path Control in Road Roundabout Environment. *International Journal of Advanced Robotic Systems* 16(2): 1-17.

Ali MAH and Mailah M (2019b) Path Planning and Control of Mobile Robot in Road Environments using Sensors Fusion and Active Force Control. *IEEE Transactions on Vehicular Technology* 68(3): 2176-2195.

Ali MAH, Mailah M, Azhar W, et al. (2015) Wheeled Mobile Robot Path Control in a Complex Trajectory using HybridMethods. In: *15th International Conference on Robotic, Control and Manufacturing (ROCOM 15*), Kualalumpur, Malaysia, 2-4April 2015, pp.38-43. USA: WSEAS

Ali MAH, Mailah M and Hing T (2012a) Path Navigation of Mobile Robot in a Road Rou-ndabout Setting. In: *1st International on Systems, Control, Power and Robotics*, Singapore 11-13 March 2012, pp. 198-203. USA: IEEE.

Ali MAH, Mailah M and Hing T (2012b) Path Planning of Mobile Robot for Autonomous Navigation of Road Roundabout Intersection. *International Journal of Mechanics* 6(4): 203-211.

Ali MAH, Mailah M, Yussof WAB, et al. (2016) Sensors Fusion based Online Mapping and Features Extraction of Mobile Robot in the Road Following and Roundabout. *IOP Conference Series: Materials Science and Engineering* 114 (1): 012135. 2016.

Boukattaya M, Damak T and Jallouli M (2011) Robust Adaptive Control for Mobile Manipulators. *International Journal of Automation and Computing 8(1):8-13.*

Chen C, Torre F and Dong W (2014) Distributed Exponentially Tracking Control of Multiple Wheeled Mobile Robots. *American Control Conference (ACC)*, Portland, USA, 4-6 June 2014, pp. 4014-4019. USA: IEEE.

Chen L and Baoli M (2015) A Nonlinear Formation Control of Wheeled Mobile Robots with Virtual Structure Approach. **In:** *2015 34th Chinese Control Conference (CCC)*, Hangzhou, China, 28-30 July 2015, pp. 1080-1085. USA: IEEE.

Chiang CK, Chung HY and Lin JJ (1997) A self-learning fuzzy logic controller using genetic algorithms with reinforcements. IEEE Transactions on Fuzzy Systems 5(3): 460 – 467.

Chiu CS, Chiang TS, and Ye YT (2015) Fuzzy Obstacle Avoidance Control of A Two-Wheeled Mobile Robot. In: *International Automatic Control Conference (CACS),* Yilan, Taiwan, 18-20 November 2015, pp.1-6. USA: IEEE.

Cui M, Liu W and Liu H et al (2016) Unscented Kalman Filter based Adaptive Tracking Control for Wheeled Mobile Robots in the Presence of Wheel Slipping. In: *12th World Congress on Intelligent Control and Automation (WCICA)*, Guilin, China, 12-15 June 2016, pp. 3335-3340. USA: IEEE.

Ghiasvand M and Alipour K (2013) Formation Control of Wheeled Mobile Robots based on Fuzzy Logic and System Dynamics. In: *13th Iranian Conference on Fuzzy Systems (IFSC)*, Qazvin, Iran, 27-29 August 2013, pp. 1-6. USA: IEEE.

Godfrey LB and Gashler MS (2017) A parameterized activation function for learning fuzzy logic operations in deep neural networks. *arXiv* (1708.08557v2): 1-6.

Guan X, Zhang P, Fang M, et al. (2014) The adaptive control for the Outdoor Mobile Robot with diameter-variable wheels. In**:** *2014 IEEE International Conference on Information and Automation (ICIA)*, Hailar, China, 28-30 July 2014, pp. 1096-1101. USA: IEEE.

Hewit JR and Burdess JS (1981) Fast Dynamic Decoupled Control for Robotics using Active Force Control. *Mechanism and Machine Theory* 16(5): 535-542.

Huang D and Zhai J (2015) Trajectory Tracking Control of Wheeled Mobile Robots Based on Disturbance Observer. In: *2015 Chinese Automation Congress (CAC)*, Wuhan, China, 27-29 November 2015, pp. 1761-1765. USA: IEEE.

Koubaa Y, Boukattaya M and Damak T (2014a) Robust control of wheeled mobile robot in presence of disturbances and uncertainties. In: *14th international conference on Sciences and Techniques of Automatic control & computer engineering - STA'2013, Sousse, Tunisia*, 20-22 Dec. 2013, 274–280. USA: IEEE.

Koubaa Y, Boukattaya M and Damak T (2014b) Adaptive Sliding-Mode Control of Nonholonomic Wheeled Mobile Robot. In: *15th international conference on Sciences and Techniques of Automatic control & computer engineering*, Hammamet, Tunisia, 21-23 Dec. 2014, pp. 336-343. USA: IEEE.

Kwek, LC, Wong EK, Loo CK, et al. (2003) Application of Active Force Control and Iterative Learning in a 5-Link Biped Robot. *Journal of Intelligent and Robotic Systems*. 2003. 37(2): 143-162.

Li Z, Wang Y, Song X, et al. (2015) Neural adaptive tracking control for wheeled mobile robots. *In: 2015 International Conference on Fluid Power and Mechatronics (FPM)*, Harbin, China, 5-7 Aug. 2015, pp. 610–617. USA: IEEE.

Liu Y, Yu S and Gao B *et al* (2015) Receding Horizon Following Control of Wheeled Mobile Robots: A Case Study. In: *IEEE International Conference on Mechatronics and Automation*, Beijing, China, 2-5 August 2015, pp. 2571–2576. USA: IEEE.

Lotfi A and Tsoi AC (1996) Learning Fuzzy Inference Systems Using an Adaptive Membership Function Scheme. IEEE Transactions on Systems. Man and Cybernetics—Part B: Cybernetics 26(2): 326-331.

Low CB and Wang D (2008) GPS-Based Tracking Control for a Car-Like Wheeled Mobile Robot With Skidding and Slipping. *IEEE/ASME transactions on mechatronics 13(4):* 480-484.

Lu X and Fei J (2015) Tracking Control of Wheeled Mobile Robots Using Iterative Learning Controller. In: *15th International Conference on Control, Automation and Systems (ICCAS 2015)*, Busan, South Korea, 13-16 Oct. 2015, pp. 1-6. USA: IEEE.

Mailah M (1998) *Intelligent Active Force Control of a Rigid Robot Arm Using Neural Network and Iterative Learning Algorithms*. PhD Thesis, University of Dundee, UK.

Mailah M, Abdullah S and Hing T (2012) Tracking Performance on Feedfoward Model Base Active Force Control of Mobile Manipulator using Matlab and ADAMS. In: *1st International Conference on Systems, Control, Power and Robotics*, Singapore, 1-2 May 2012, pp.83-88. USA: WSEAS

Mu J, Yan XG, Spurgeon SK (2015) Trajectory Tracking Control of a Two-Wheeled Mobile Robot Using Sliding Mode Techniques. In: *34th Chinese Control Conference*, Hangzhou, China, 28-30 July 2015, pp. 3307-3312. USA: IEEE.

Partovibakhsh M and Liu G (2015) Slip ratio estimation and control of wheeled mobile robot on different terrains. In: *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, Shenyang, China, 8-12 June 2015, pp. 566-571. USA: IEEE.

Pitowarno E, Mailah M and Jamaluddin H (2003) Knowledge-based Trajectory Error Pattern Method Applied to an Active Force Control Scheme. *IIUM Engineering Journal* 3(1): 1-15.

Raeisi Y, Shojaei K and Chatraei A (2015) Output feedback trajectory tracking control of Car-like Drive Wheeled Mobile Robot Using RBF Neural Network. In: *The 6th Power Electronics, Drive Systems & Technologies Conference (PEDSTC2015)* , Tehran, Iran, 3-4 February 2015, pp. 363-368. USA: IEEE.

Sabzehmeidani Y, Mailah M and Hussein M (2011) Modelling and control of a piezo actuated micro robot with active force control capability for in-pipe application. *Int. J. Modelling, Identification and Control 13(4):301-308.*

Song Z, Ren H, Zhang J, etal. (2016) Kinematic Analysis and Motion Control of Wheeled Mobile Robots in Cylindrical Workspaces. *IEEE transactions on automation science and engineering 13(2):* 1207 – 1214.

White Drive Products (2005) Drive Wheel Motor Torque Calculations. USA. 2005

Yang, H., Fan, X., Shi, P., & Hua, C. (2016). Nonlinear Control for Tracking and Obstacle Avoidance of a Wheeled Mobile Robot With Nonholonomic Constraint. In: *IEEE Transactions on Control Systems Technology*, *24*(2), 741–746.

Yoon J, Oh JH, Park JH, *et al* (2014) Autonomous Dynamic Driving Control of Wheeled Mobile Robots. In: *IEEE International Conference on Robotics & Automation (ICRA),* Hong Kong, China, 31 May-7 June 2014, pp. 5274-5279. USA: IEEE.

Zidani G, Drid S, Chrifialaoui L, *et al* (2014) Nonlinear Tracking Control of a Wheeled Mobile Robot. In: *15th International Conference On Sciences and Techniques of Automatic Control & Computer Engineering*, Hammamet, Tunisia, 21-23 December 2014, pp.325-331. USA: IEEE**.**