

CONVENTIONAL FUZZY LOGIC CONTROLLER FOR BALANCING TWO-WHEEL INVERTED PENDULUM

Doaa M. Abd-elaziz

Engdoam92@gmail.com

Abstract

This paper presents the design and the Real-Time implementation of self-balancing Two-Wheeled Inverted Pendulum (TWIP) using state-feedback controller and a conventional fuzzy logic controller (CFLC) on Real-Time. The state-feedback controller consists of two parts PD controller and PI controller. The state-feedback controller was designed first for the nonlinear model then it was updated for the Real-Time implementation. The CFLC was designed first based on the state-feedback controller to reach the point of using pure fuzzy controller for the TWIP. The CFLC was designed with Mamdani fuzzy inference architecture if-then rules. The TWIP is balanced with CFLC with a base of 7 rules only. Different controller's types were tested on real-Time to perform a good balance for the TWIP. Fuzzy PI, Fuzzy PD, and combination of two independent CFLC were used to achieve the balancing of the TWIP on Real-Time.

Keywords—*State-Feedback controller, Fuzzy Logic controller, Fuzzy PI, Fuzzy PD, Two-wheeled Inverted Pendulum, Mamdani FIS, Real-Time Control.*

I. Introduction

The TWIP system is a very popular and famous benchmark for testing dynamics and implementing different control techniques. The TWIP system has attracted many researchers due to two main reasons, firstly due to the wide spread use of this system and, secondly because it represents a challenge problem from control design perspective. The basic challenging task about TWIP system is to balance on the upright position without falling. Researchers improved different control techniques for controlling and investigating the behavior of the TWIP, like Proportional-Integral-Derivative (PID), Linear-Quadratic-Regulator (LQR), Neural Networks (NN), Fuzzy Logic (FL), Genetic Algorithms (GA), and Adaptive Neuro-Fuzzy inference system (ANFIS). The main aim of our task was to self-balance the TWIP robot in the Real-Time. For this purpose we designed and implemented fuzzy logic controller (FLC). Fuzzy controller is an intelligent control technique to represent and perform the human knowledge about how to control a system to make control decisions. Many researches on fuzzy logic control have been investigated for the TWIP system either

only with fuzzy controller or combination of other controller.

Cheng-Haoet al. [1] designed and implemented a control scheme with three FLC for balancing the TWIP. Nasir et al. [2] compared the performance of a FLC with PID for balancing the TWIP. Akmal et al. [3] used FLC to stabilize the two wheeled EV3 LEGO robot. Xu et al. [4] implemented and validated a real time FLC based on Takagi-Sugeno using 16 fuzzy rules to achieve position control of the wheel. Junfeng et al. [5] designed a state-feedback controller using an approximate linear state-space model. Then, they used MATLAB software to compare this controller with a fuzzy logic controller. Yong et al. [6] designed fuzzy controller based on the T-S fuzzy model with the parallel distribution compensator (PDC) for balancing the TWIP. Chiu and Chang [7] proposed and derived a simple Mamdani-like fuzzy controller for self-balancing wheeled human-conveyance vehicle (WHCV), which it is a special type of TWIP. Qingcheng and Jian [8] designed a fuzzy immune PD controller to achieve self-balance of TWIP and compared its performance with those of conventional fuzzy PD controller. Sadeghian and Masoulethey [9] designed three

controllers to balance the TWIP, PD controller, PID controller and Fuzzy-PID. Controller's parameters were tuned with the Genetic Algorithm and fuzzy logic. Huang et al. [10] obtained the desired position and direction while keeping the TWIP balanced by integrating four interval type-2 fuzzy logic systems (IT2 FLSs). Sayidmarie et al. [11] designed FLC for balancing TWIP by considering the tilt angle as feedback and coded and stored the fuzzy rules in a microcontroller memory. Hao et al. [12] designed FLC for balancing the TWIP and studied the angle, angular velocity and displacement using simulation.

Based on this, controlling and balancing the TWIP is a challenging task. The aim of the work is to control and balance the TWIP using fuzzy logic controllers. The work presented in this paper is a basic necessary for studying the behavior of the TWIP using fuzzy logic controller to understand how the system responses, so an advanced techniques and different types of fuzzy logic controllers will be designed in the near future. Three types of fuzzy controllers will be designed and tested in real-time to balance the TWIP robot.

This paper is organized as follows: Overview of TWIP system is described in section II. State-feedback controller design is represented in section III. Fuzzy logic basics are discussed in section IV. Different

conventional fuzzy controller's designs are represented in section V. Real-Time results for balancing the TWIP robot are in section VI. Section VII summarizes the conclusion.

II. Description of the TWIP system

A. Hardware Description

The TWIP is a system with two geared DC motors coupled with two wheels, which is mounted together on the mechanical chassis. Figure 1 shows the overview of the TWIP system. To measure the pendulum tilt angle and the angular velocity, An Magnetic Pickup Unit (MPU) is used, which is equipped with an accelerometer and a gyroscope. Two geared DC motors with encoders generate the required torque to drive the wheels. Motor encoders readings provide the displacement and speed of the TWIP. Arduino UNO is used as an input /output (I/O) data acquisition card (DAQ). The controller is implemented in MATLAB Simulink Desktop Real-Time to provide an easy way to manage the system externally using personal computer (PC). The controller takes the system states as an input and provides a control signal to generate a pulse-width-modulation (PWM) signal for the motors. For our research, we used a commercial robot for the TWIP called Balanbot as shown in Figure 2.

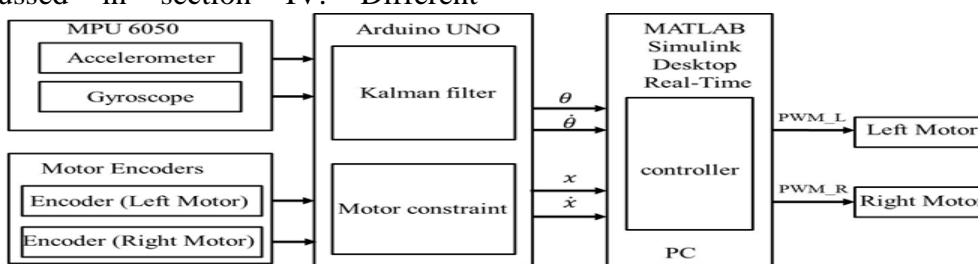


Fig.1 system overview for control of the TWIP.

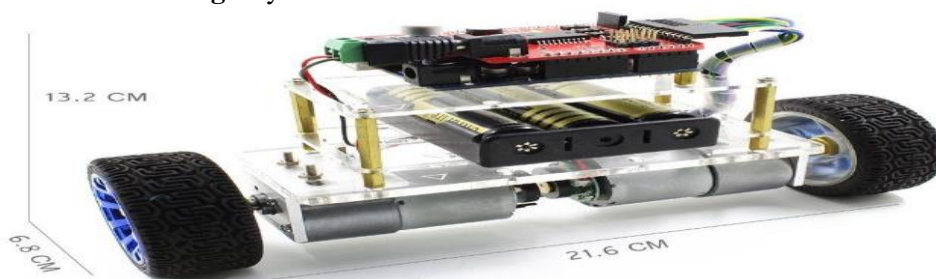


Fig.2 Balanbot robot

B. Modeling

The dynamics of the robot has to be described by a mathematical model for an efficient control system , so the equations of motion for the TWIP and linear model for a DC motor is derived in details in [13],and nonlinear equations of the TWIP are:

$$(I_p + M_p L^2) \ddot{\theta}_p - \frac{2K_m K_e}{Rr} \dot{x} + \frac{2K_m}{R} V_a + M_p g L \sin(\theta_p) = -M_p L \ddot{x} \cos(\theta_p) \quad (1)$$

$$\frac{2K_m}{Rr} V_a = \left(2M_w + \frac{2I_w}{r^2} + M_p \right) \ddot{x} + \frac{2K_m K_e}{Rr^2} \dot{x} + M_p L \ddot{\theta}_p \cos(\theta_p) - M_p L \dot{\theta}_p^2 \sin(\theta_p) \quad (2)$$

and parameter sets are as:

Mw: Wheel weight

Mp: Body weight

Iw: Wheel inertia

Ip: Body inertia

L:Distance between the centers of the wheel and the robot's center of gravity (COG distance)

r:Wheel radius

R:Motor resistance

K_m, K_e : Constant for motor torque/EMF

g:Acceleration due to gravity

V_a :Input

X: Position

θ : Tilt angle

Due to using a commercial robot for the research, the robot parameters needed to be estimated. Parameter estimation process was done [internal report] and the final estimated parameters are shown in Table 1.

Table (1) TWIP Parameters.

TWIP Parameters	Symb ol	value
Wheel inertia	Iw	0.99987
Body inertia	Ip	0.035441
Body weight	Mp	0.8408
Wheel weight	Mw	0.068524
COG distance	L	0.058984
Wheel radius	r	0.029
Motor resistance	R	4.3132
Constant for motor torque/EMF	K_m, K_e	0.60778
Acceleration due to gravity	G	9.8

Equations (1) and (2) can be linearized by assuming $\theta_p = \pi + \varphi$, where φ represents a small angle. Therefore,

$$\cos(\theta_p) = -1, \sin(\theta_p) = -\varphi \text{ and } \left(\frac{d\theta_p}{dt} \right)^2 = 0$$

The rearranged linearized equations are,

$$\ddot{\varphi} = \frac{M_p L}{(I_p + M_p L^2)} \ddot{x} + \frac{2K_m K_e}{Rr(I_p + M_p L^2)} \dot{x} - \frac{2K_m}{R(I_p + M_p L^2)} V_a + \frac{M_p g L}{(I_p + M_p L^2)} \varphi \quad (3)$$

$$\ddot{x} = \frac{2K_m}{Rr(2M_w + \frac{2I_w}{r^2} + M_p)} V_a - \frac{2K_m K_e}{Rr^2(2M_w + \frac{2I_w}{r^2} + M_p)} \dot{x} + \frac{M_p L}{(2M_w + \frac{2I_w}{r^2} + M_p)} \ddot{\varphi} \quad (4)$$

The state-space representation is,

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\varphi} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2K_m K_e (M_p L r - I_p - M_p L^2)}{Rr^2 \alpha} & \frac{M_p^2 g L^2}{\alpha} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2K_m K_e (r\beta - M_p L)}{Rr^2 \alpha} & \frac{M_p g L \beta}{\alpha} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \varphi \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2K_m (I_p + M_p L^2 - M_p L r)}{Rr \alpha} \\ 0 \\ \frac{2K_m (M_p L - r\beta)}{Rr \alpha} \end{bmatrix} V_a \quad (5)$$

$$\begin{bmatrix} x \\ \varphi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} V_a \quad (6)$$

where,

$$\beta = \left(2M_w + \frac{2I_w}{r^2} + M_p \right)$$

$$\alpha = (I_p \beta + 2M_p L^2 (M_w + \frac{I_w}{r^2}))$$

Using the parameters from Table 1 into (5) the state-space model is,

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\varphi} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.08241 & 0.0002641 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 153.8 & 12.67 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \varphi \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.003932 \\ 0 \\ -7.341 \end{bmatrix} V_a$$

The open loop transfer functions for x and φ from input V_a to output are,

$$x = \frac{0.003932s^2 - 0.05175}{s^4 + 0.08241s^3 - 12.67s^2 - 1.085s} \quad (7)$$

$$\varphi = \frac{-7.341s}{s^3 + 0.08241s^2 - 12.67s - 1.085} \quad (8)$$

III. State-feedback controller

For designing and testing fuzzy

controller for the TWIP robot, we needed to understand the behavior of the robot so a state-feedback controller was designed. The state-feedback controller was used to balance and observe the behavior of the TWIP robot. The

state-feedback controller divided into two parts, first part is a PD controller for angle, and second part is a PI controller for speed. Figure 3 shows the state-feedback controller of the TWIP robot.

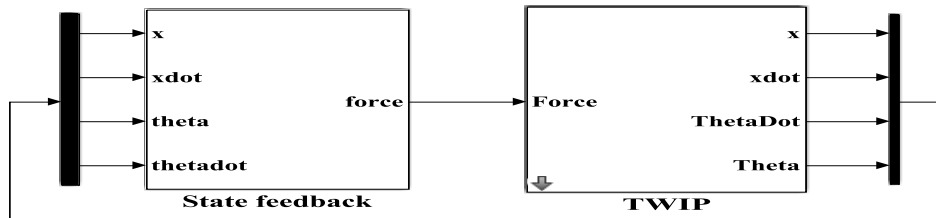


Fig.3 State-feedback controller.

For obtaining the controller's gains for PD controller and PI controller, first the closed-loop transfer functions were calculated for both systems shown in Figures 4 and 5.

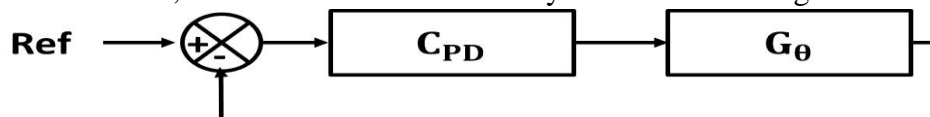


Fig.4 Feedback system for angle.

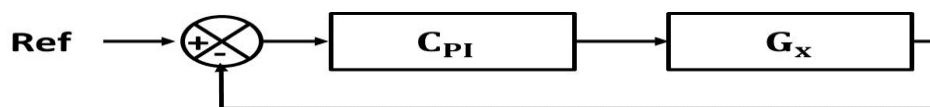


Fig.5 Feedback system for speed.

For PD controller of angle,

$$C_{PD} = K_p + K_d s \quad (9)$$

The closed-loop transfer function,

$$G_{\theta c} = \frac{-7.341s(K_p + K_d s)}{s^3 + (0.08241 - 7.341K_d)s^2 + (12.67 - 7.341K_p)s - 1.085} \quad (10)$$

For PI controller of speed,

$$C_{PI} = K_p + \frac{K_i}{s} \quad (11)$$

The closed-loop transfer function,

$$G_{xc} = \frac{(0.003932s^2 - 0.05175)(K_p + K_d s)}{s^5 + 0.08241s^4 + (0.003932K_p - 12.67)s^3 + (0.003932K_i - 1.085)s^2 + (-0.0517K_p)s - 0.05175K_i} \quad (12)$$

Parameters of the controllers can be determined by modeling process dynamics and applying some method for control design. Since the complexity of the controller is directly related to the complexity of the model it is necessary to have models of low order. From equations 10 and 12 both of the characteristic equations for the TWIP robot reveal that it is hard to tune the parameters of each controller using classical or theoretical methods.

The gains values and methods to choose the parameter of the controllers depend largely on

the characteristics of system. A method producing good result for a system may not work at all for another system with different characteristics. Theoretically, the PD and PI values depend on the state of the system, such as Mechanical Structure, Physical properties, Electrical properties. But practically, it also depends on the external conditions, like Atmospheric conditions.

The TWIP is a highly nonlinear system, due to that There is no clear boundary for taking P, I or D values and is mostly taken based on experience. In case of the TWIP robot before

adjusting the gains values, we need to know what P, I, and D values means practically. P gain, determines the force which the robot will correct, a lower P shows the disability of balancing itself and a higher P shows the aggressive behavior. I gain, determines the response time the robot take to correct itself. D gain, smooth and depress the robot oscillations, a lower D is unable to remove oscillations and a higher D will cause aggressive vibrations.

The simple algorithm for practically tuning the gains is,

```

Start
Set I gain and D gain to 0.
Increase P gain.
If the robot starts to oscillate (move) about
the balance position.
Then set the P gain to that value.
Else increase the P gain.
With P set, increase I gain.
If the robot accelerates faster.
Then set the I gain to that value.
Else increase the I gain.
With P and I set, increase D.
If the robot moves about its balanced
position smoother.
Then set the D gain to that value
Else increase D gain
End
    
```

The state-feedback controller's parameters that achieved balancing were, for PD controller the gains are $K_p=25$ and $K_d=3.5$ and for PI controller $K_p=30$ and $K_i=0.34$.

IV. Fuzzy Logic

L.A. Zadeh founded fuzzy sets in 1965. In 1974 E.H. Mamdani utilized the fuzzy theory and put it in practice when he implemented a FLC to a steam engine. Other practical applications of FLC were invented, like the control of the Subway in Japan in 1984, automated aircraft vehicle landing in 1987, and in 1990 Sony company present first TV using fuzzy. Since then, many researches led to rapid developments in fuzzy.

Fuzzy is a computational model based on humans reasoning. In fuzzy controllers the underlying mechanics are presented using linguistics rather than mathematics. Figure 6 shows a typical FLC contains four parts: fuzzifier, rule-base, inference engine, and defuzzifier.

The fuzzifier converts the crisp input to a linguistic variable. The inference, converts the input to the output using If-Then fuzzy rules. The defuzzifier converts the output of the inference engine to crisp. Fuzzy knowledge base stores the fuzzy rules and membership functions.

The expression convention indicates the basic concept of building FLC as a multi-input controller. Many researches have been done to develop the FLC, and new methods for designing FLC appeared. Researchers replaced the conventional multi-input FLC with single input FLC for better performance. In this paper we used the conventional FLC to balance the TWIP robot.

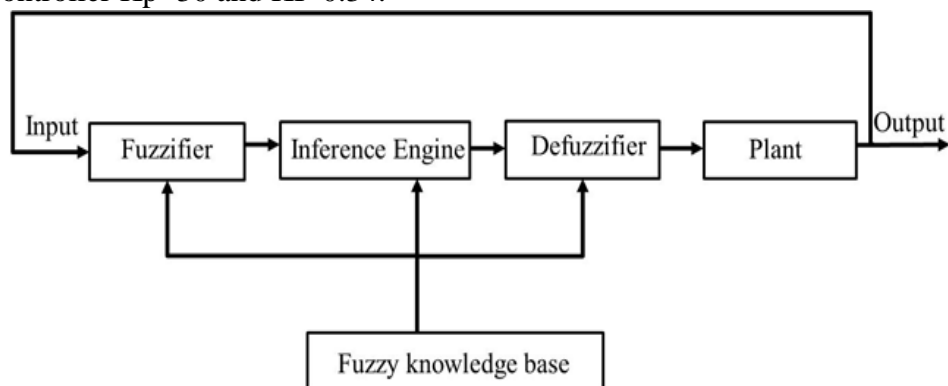


Fig.6 Fuzzy control system.

V. Conventional Fuzzy controllers

To balance the TWIP in upright position using CFLC, the shape and number of membership functions, the number of the fuzzy rules, and the inference type needed to be chosen wisely to provide the ability of implementing and testing the controller in Real-Time. The triangular shape of membership is faster than Gaussian shape membership. The less number of memberships means less fuzzy rules, which lead to fast computations which is needed in Real-Time to achieve good stability. This paper is focused on only the balancing of the TWIP then, we don't need all the fuzzy rules on the fuzzy matrix rules. We can only use those rules that serve the purpose of balancing the TWIP. This technique of using only the rules that serve the needed task was used by different researchers as in [3] and [14]. The Mamdani inference type is suitable for controllers that work on Real-Time, unstable systems, and need less computation's time to guarantee fast output.

Based on that, Table 2 show only the balancing rules of the TWIP, Those inputs and output share same shape of membership which is triangular and linguistics variables are NM is negative medium, NS is negative small, ZO is zero, PS is positive small, and PM is positive medium.

Table (2) the balancing rules for the TWIP

θ/x	$\dot{\theta}/\dot{x}$				
	NM	NS	ZO	PS	PM
NM			NM		
NS		NS		ZO	
ZO			ZO		
PS		ZO		PS	
PM			PM		

The fuzzy rules for balancing the TWIP in table 2 are used in the designing process in the form of IF-THEN rules:

- IF θ/x is NM and $\dot{\theta}/\dot{x}$ is ZO THEN U is NM
- IF θ/x is NS and $\dot{\theta}/\dot{x}$ is NS THEN U is NS
- IF θ/x is NS and $\dot{\theta}/\dot{x}$ is PS THEN U is ZO
- IF θ/x is ZO and $\dot{\theta}/\dot{x}$ is ZO THEN U is ZO
- IF θ/x is PS and $\dot{\theta}/\dot{x}$ is NS THEN U is ZO
- IF θ/x is PS and $\dot{\theta}/\dot{x}$ is PS THEN U is PS
- IF θ/x is PM and $\dot{\theta}/\dot{x}$ is ZO THEN U is PM

To balance the TWIP robot using fuzzy controller this was done step-by-step. First, the two parts of the state-feedback controller were used separately with fuzzy controller for the Fuzzy PD controller and the Fuzzy PI controller then they were eliminated to use a pure fuzzy controller for balancing the TWIP robot for the two independent fuzzy controllers.

A. Fuzzy PD

Fuzzy PD controller is CFLC for controlling both of the displacement and the speed of the TWIP robot which is combined with PD controller for the tilt angle and the rate of the tilt angle. Figure 7 shows the block diagram of the Fuzzy PD controller.

The FLC in this type has two inputs and one output, the inputs are the displacement (x) of the TWIP with range [-600,600], and the speed (x.) with range [-2, 2], the output of the FLC is the control signal based on the decisions as designed in the rule-base with range [-1000, 1000]. Figures 8-11 show the membership shape and ranges for the inputs and output and the fuzzy rules for balancing the TWIP.

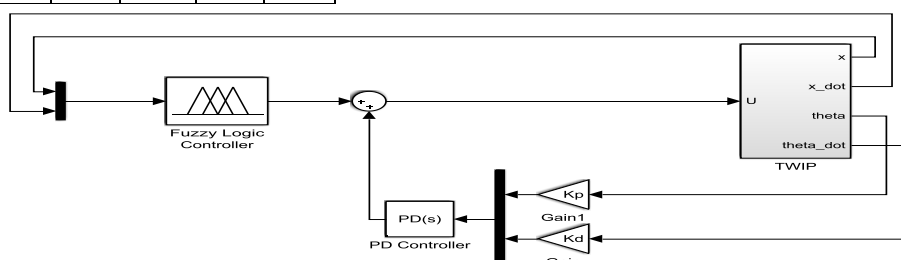


Fig.7 Fuzzy PD controller

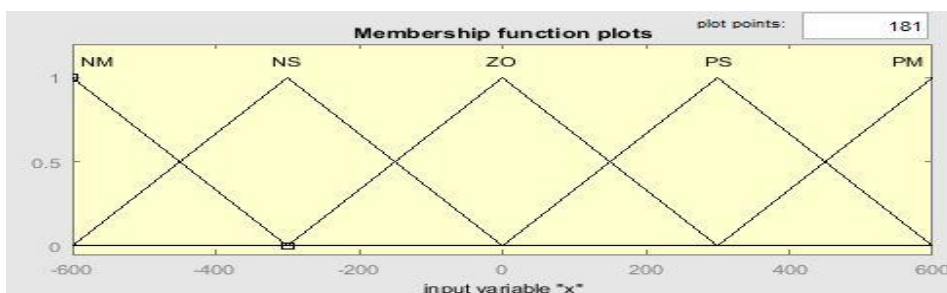


Fig.8 Membership function for x.

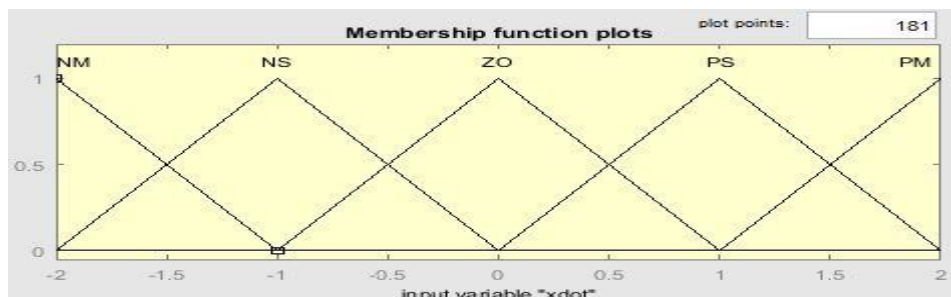


Fig.9 Membership function for x_dot



Fig.10 Membership function for the control output.

1. If (x is PM) and (xdot is ZO) then (xu is PM) (1)
 2. If (x is PS) and (xdot is PS) then (xu is PS) (1)
 3. If (x is PS) and (xdot is NS) then (xu is ZO) (1)
 4. If (x is NM) and (xdot is ZO) then (xu is NM) (1)
 5. If (x is NS) and (xdot is NS) then (xu is NS) (1)
 6. If (x is NS) and (xdot is PS) then (xu is ZO) (1)
 7. If (x is ZO) and (xdot is ZO) then (xu is ZO) (1)

If x is: NM, NS, ZO, PS, PM, none
 and xdot is: NM, NS, ZO, PS, PM, none
 Then xu is: NM, NS, ZO, PS, PM, none

Connection: or and
 Weight: 1
 Buttons: Delete rule, Add rule, Change rule, <<, >>

FIS Name: FLC_POS_7RULES
 Help, Close

Fig.11 Fuzzy rules for the displacement and the speed of the TWIP.

B. Fuzzy PI

Fuzzy PI controller is CFLC for both of the angle and the rate of the angle which is combined with PI controller for the displacement and the speed of the TWIP robot. Figure 12 shows the block diagram of the Fuzzy PI controller.

The FLC in this type has two inputs and one output, the inputs are the tilt angle (θ) of the TWIP with range $[-100,100]$, and the tilt rate ($\dot{\theta}$) with range $[-250,250]$, the output of the FLC is the control signal based on the

decisions as designed in the rule-base with range $[-2000, 2000]$. Figures 13-16 show the membership shape and ranges for the inputs and output and the fuzzy rules for balancing the TWIP.

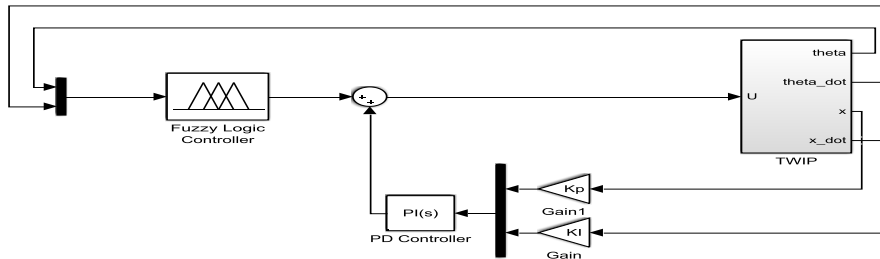


Fig.12 Fuzzy PI controller.

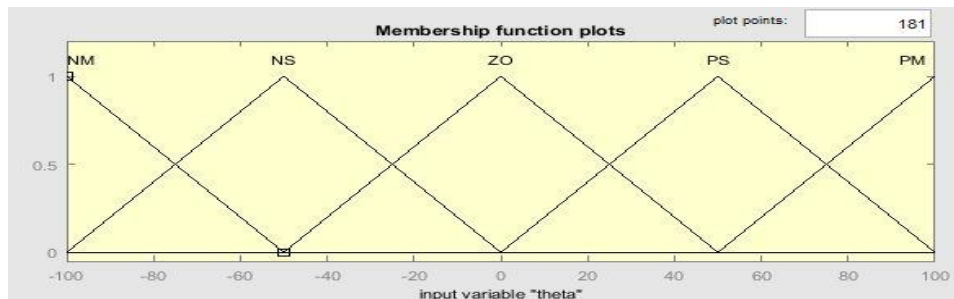


Fig.13 Membership function for theta.

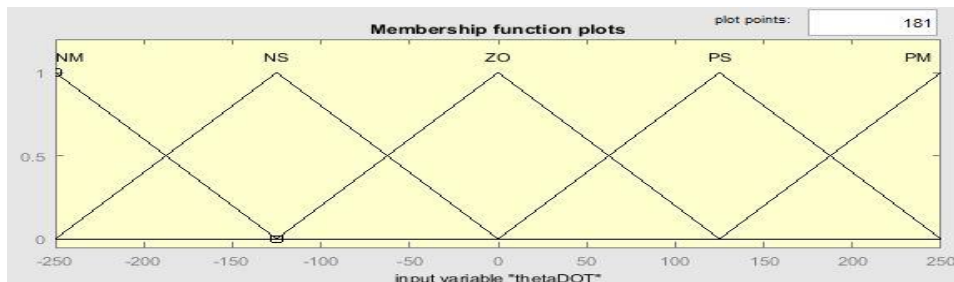


Fig.14 Membership function for theta_dot.

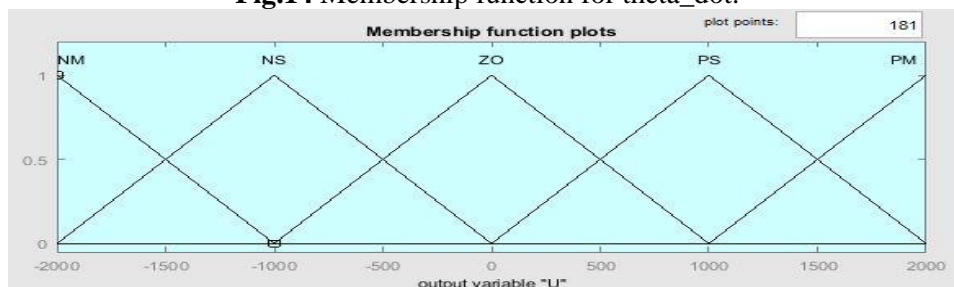


Fig.15 Membership function for the control output.

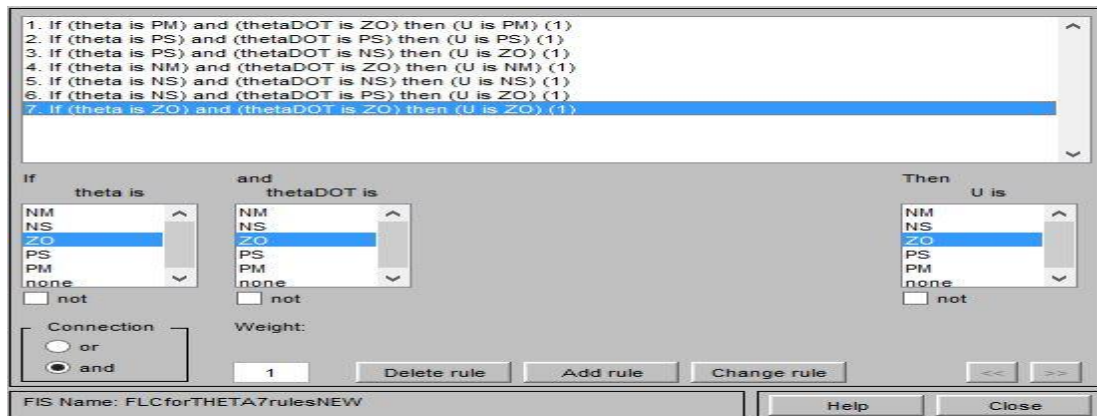


Fig.16 Fuzzy rules for the tilt angle and the rate of tilt angle of the TWIP.

C. Two independent CFLC

Two independent CFLC were implemented, one for the angle and the other for the displacement and the speed. In this type we combined the above two fuzzy logic controllers to balance the TWIP to observe the balancing with only pure FLC as shown in Figure 17.

The first CFLC has two inputs and one output, the inputs are the angle (θ) of the TWIP with range [-100,100], and the rate ($\dot{\theta}$)

with range [-250,250], the output of the FLC is the control signal based on the decisions as designed in the rule-base with range [-2000, 2000].

The second CFLC has two inputs and one output, the inputs are the displacement (x) of the TWIP with range [-600,600], and the speed (\dot{x}) with range [-2, 2], the output of the FLC is the control signal based on the decisions as designed in the rule-base with range [-1000, 1000].

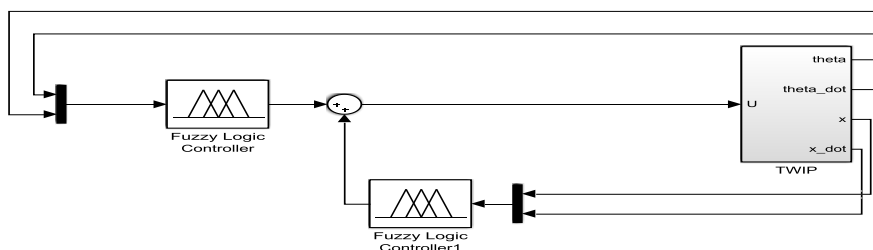


Fig.17 two independent conventional fuzzy logic controllers.

VI. Real-Time results

All controllers were tested and run on the Real-Time to observe the different behavior of each controller in achieving the purpose of balancing the TWIP robot.

A. State-feedback controller

The state-feedback controller balances the angle of the TWIP robot as shown in Figure 18.

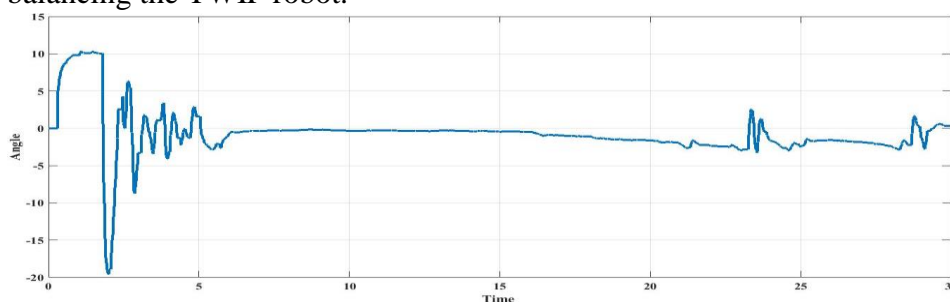


Fig.18 Angle of the TWIP robot for the State-feedback controller.

B. Fuzzy PD

The fuzzy PD controller tries to balance the TWIP robot in upright position so; it moves the robot forward and backward until it

balances itself. After few seconds the TWIP robot stay in the balance mood if no external force applied as shown in Figure 19.

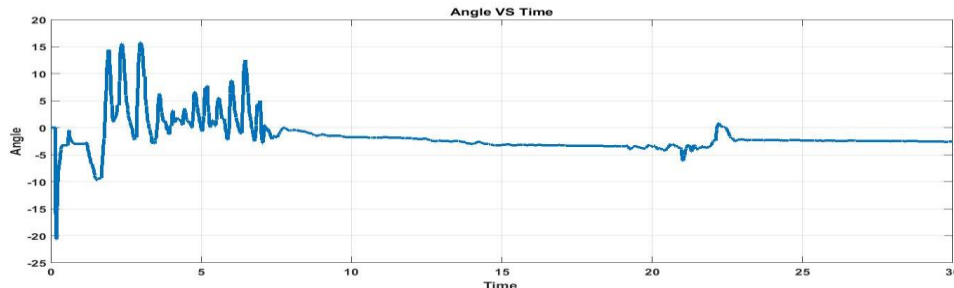


Fig.19 Angle of the TWIP robot for the Fuzzy PD controller.

C. Fuzzy PI

The fuzzy PI controller tries to balance the TWIP robot in upright position so; it moves the robot forward and backward to try to enter the balance mood but, the TWIP robot enters the balance mood for few seconds and then start to move itself again. This movement of The TWIP robot that shows it doesn't stay

in the balance mood is due to external disturbance. Disturbance here is due to the environment such as wind, air, fan, and even if someone move around. So it is a good controller and can resist and rebalance itself if any external force is applied as shown in Figure 20.

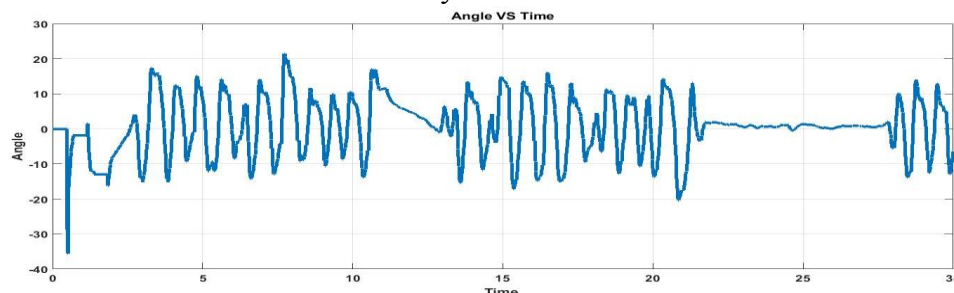


Fig.20 Angle of the TWIP robot for the Fuzzy PI controller.

D. Two Independent CFLC

The two independent fuzzy logic controllers try to balance the TWIP robot in upright position so; it moves the robot forward and backward to try to enter the balance mood if there is an external disturbance. Those two pure fuzzy logic controllers make the TWIP

robot move very close to the balance mood due to the time it takes to make two decisions for both the two fuzzy controllers, it is a good controller and can adapt with any applied external force and rebalance itself as shown in Figure 21.

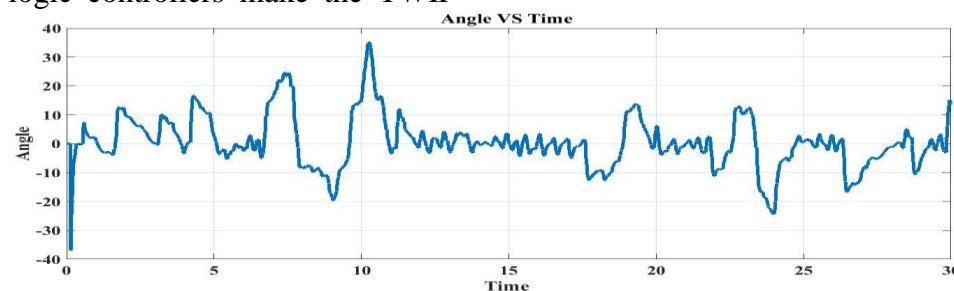


Fig.21 Angle of the TWIP robot for the two independent fuzzy logic controllers.

-figure 22 showing the response of the three fuzzy controllers used for the TWIP system

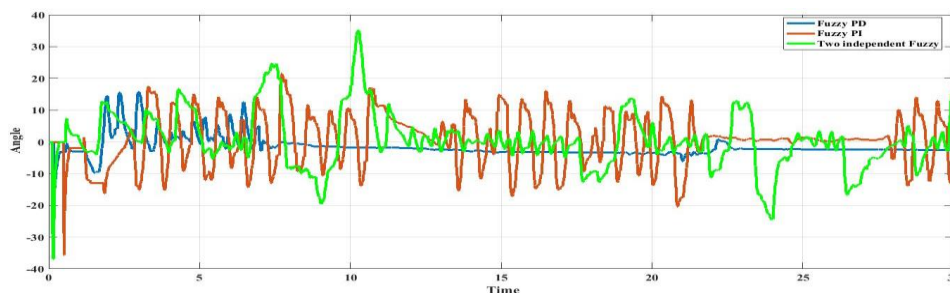


Fig.22 the angle of the TWIP robot for the fuzzy controllers.

VII. Conclusion

In this paper we used a state-feedback controller to balance and study the behavior of the TWIP robot. The conventional fuzzy logic controller was used to balance the TWIP robot. We used different types of fuzzy logic controllers to study the behavior of changing the type of controller on the balancing of the TWIP robot. We implement and test all types provided in this paper in the Real-Time. In the fuzzy PD, the TWIP robot at the beginning tried to balance itself and then stay in the balance mood. The fuzzy PI, the TWIP robot at the beginning tried to balance itself and then stay in the balance mood for a while then start to balance itself again and then stay in the balance mood and so on. The two independent fuzzy logic controllers, the TWIP robot always move very close around the balance values trying to stay in the balance mood but it doesn't it still move around. All the three controllers showed a good controller in the real-time the TWIP robot always balanced and never falling. All the three fuzzy showed a good control response when applying external force.

References

- [1] H. Cheng-Hao, W.-J. Wang, and C. Chih-Hui, "Design and Implementation of Fuzzy Control on a Two-Wheel Inverted Pendulum," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 7 pp. 2988-3001, July 2011.
- [2] A. N. K. Nasir, M. A. Ahmad, R. Ghazali, and N. S. Pakheri, "Performance Comparison between Fuzzy Logic Controller (FLC) and PID Controller for a Highly Nonlinear Two-Wheels Balancing Robot," *IEEE Conference on Informatics and Computational Intelligence (ICI)*, pp. 176-181, December 2011.
- [3] M. A. Akmal, N. F. Jamin, and N. M. Abdul Ghani, "Fuzzy logic controller for two wheeled EV3 LEGO robot," *IEEE Conference on Systems, Process and Control (ICSPC)*, pp. 134-139, December 2017.
- [4] J. Xu, Z.-Q. Guo, and T. H. Lee, "Design and Implementation of a Takagi Sugeno-Type Fuzzy Logic Controller on a Two-Wheeled Mobile Robot," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5717-5728, 2013.
- [5] J. Wu and W. Zhang, "Design of fuzzy logic controller for two-wheeled self-balancing robot," *IEEE Conference on Strategic Technology*, pp. 1266-1270, August 2011.
- [6] Q. Yong, L. Yanlong, Z. Xizhe, and L. Ji, "Balance control of two-wheeled self-balancing mobile robot based on TS fuzzy model," *IEEE Conference on Strategic Technology*, pp. 406-409, August 2011.
- [7] C.-H. Chiu, C.-C. Chang, "Design and Development of Mamdani-Like Fuzzy Control Algorithm for a Wheeled Human-Conveyance Vehicle Control," *IEEE Transactions on Industrial*

- [8] Electronics, vol. 59, no. 12, pp. 1814-1822, December 2012.
- [9] W. Qingcheng and F. Jian, "Fuzzy Immune PD Algorithm Applied in the Self-Balancing Two-Wheeled Robot," IEEE Conference on Future Generation Communication and Networking, pp. 112–115, December 2014.
- [10]R. Sadeghian and M. Tale Masoule, "An experimental study on the PID and Fuzzy-PID controllers on a designed two-wheeled self-balancing autonomous robot," IEEE Conference on Control, Instrumentation, and Automation (ICCIA), pp. 313–318, January 2016.
- [11]J. Huang, M-H. Ri, D. Wu, and S-H. Ri, "Interval Type-2 Fuzzy Logic Modeling and Control of a Mobile Two-Wheeled Inverted Pendulum," IEEE Transactions on Fuzzy Systems, October 2017.
- [12]O. K. Sayidmarie, M. O. Tokhi, A. M. Almeshal, and S. A. Agouri, "Design and real-time implementation of a fuzzy logic control system for a two-wheeled robot," IEEE Conference on Methods & Models in Automation & Robotics (MMAR), pp. 569–572, August 2012.
- [13]Q. Hao, P. Li, Y- Z. Chang, and F. Yang, "The fuzzy controller designing of the self-balancing robot," IEEE Conference on Electronics and Optoelectronics, pp. 16–19, July 2011.
- [14]Ooi R. Chi, "Balancing a Two-Wheeled Autonomous Robot", University of Western Australia, School of Mechanical Engineering, Final Thesis 2003.
- [15]S. Omatu; T. Ide, "Stabilization of inverted pendulum by neuro-control," IEEE Conference on Computational Intelligence, pp 2367-2372, June 1994.

المتحكم المنطقي الضبابي التقليدي لموازنة البندول المعكوس المتزن على عجلتين

دعاء محمود

ملخص البحث:

الهدف من هذا البحث هو تصميم و تطبيق عملي باستخدام المتحكم المنطقي الضبابي على البندول المعكوس المتزن على عجلتين. اولاً تم تصميم و تطبيق عملي متحكم يعتمد على حالات التغذية الراجعة للنظام ليعمل على موازنة الانسان الالي للبندول المعكوس المتزن على عجلتين و ايضا ليكون مساعد في عملية تصميم و تطبيق المتحكم المنطقي الضبابي خطوة بخطوة. تم تصميم المتحكم المنطقي الضبابي باستخدام بنية العالم ابراهيم مامداني عن طريق قواعد اذا-اذن. الانسان الالي للبندول المعكوس تم موازنته باستخدام المتحكم المنطقي الضبابي باستخدام ٧ قواعد فقط. انواع مختلفة من المتحكمات المعتمدة في تصميمها على المتحكم المنطقي الضبابي تم اختبارها و تطبيقها عمليا لتحقيق موازنة ذاتية للانسان الالي للبندول المعكوس المتزن على عجلتين.