MUĞLA

Mugla Journal of Science and Technology

DESIGN OF A MODEL REFERENCE ADAPTIVE PID CONTROLLER FOR DC MOTOR POSITION CONTROL: COMPARED WITH PID AND FUZZY CONTROLLERS

Yasin ÇAKAR, Mechatronic Engineering, Kocaeli University, Turkey, yasincakar.mech@gmail.com
(https://orcid.org/0000-0001-8021-257X)
Yıldıray ORMAN, Mechatronic Engineering, Kocaeli University, Turkey, yildirayorman@gmail.com
(https://orcid.org/0000-0002-2544-2935)
*Selçuk KİZİR, Mechatronic Engineering, Kocaeli University, Turkey, selcuk.kizir@kocaeli.edu.tr
(https://orcid.org/0000-0002-0582-5904)

Received: 31.12.2019, Accepted: 14.06.2020 Research Article
*Corresponding author
DOI: 10.22531/muglajsci.668552

Abstract

Sometimes conventional feedback controllers may not perform well due to changes in environmental conditions, changes in process dynamics that occur over time and changes in characteristics of disturbances. To overcome these problems, adaptive control methods have been developed. In this regard, a Model Reference Adaptive PID Controller (MRAPIDC) is designed using the MIT rule for position control of the DC motor in this study. At the same time, this control method is implemented using a low cost STM32F4 application development kit with Matlab-Simulink supported Waijung block set and compared with PID and fuzzy logic control methods. The results obtained with/without measurement noise disturbance under unit step and sinusoidal inputs are presented.

Keywords: Model Reference Adaptive PID, Real-Time, Waijung Blok Set, DC Motor Position Control, MIT Rule, STM32f4

DC MOTOR KONUM KONTROLÜ İÇİN MODEL REFERANS UYARLAMALI PID DENETLEYİCİ TASARIMI: PID VE BULANIK DENETLEYİCİLERİYLE KARŞILAŞTIRILMASI

Özet

Bazen geleneksel geri beslemeli kontrolörleri, çevresel koşullarda oluşan değişikler, zamanla oluşan proses dinamiklerindeki değişiklik ve bozucu etkenlerin karakteristiklerindeki değişimlerden dolayı iyi performans göstermeyebilir. Bu sorunların üstesinden gelebilmek için, uyarlamalı denetim yöntemleri geliştirilmiştir. Bu bağlamda, DC motorun konum kontrolü için MIT kuralı kullanarak Model Referans Uyarlamalı PID Denetleyici (MRUPIDD) tasarlanarak uygulanmıştır. Aynı zamanda, bu kontrol yöntemi Matlab-Simulink destekli Waijung blok seti ile düşük maliyetli STM32F4 uygulama geliştirme kiti kullanılarak gerçeklenmiş ve PID ve bulanık mantık kontrol yöntemleri ile karşılaştırılmıştır. Birim basamak ve sinüzoidal girişler ve ölçüm gürültüsü gibi bozucu etkiler altında elde edilen sonuçlar sunulmuştur.

Anahtar Kelimeler: Model Referans Uyarlamalı PID, Gerçek Zamanlı, Waijung Blok Set, Dc Motor Konum Kontrol, MIT Kuralı, STM32f4

Cite

Çakar, Y., Orman, Y., Kizir, S., (2020). "Design of a model reference adaptive pid controller for dc motor position control: compared with pid and fuzzy controllers", Mugla Journal of Science and Technology, 6(0), 25-35.

1. Introduction

One of the purposes of a closed loop control system is to eliminate the disturbances from external factors and to follow the reference signal by reducing the error [1].

More than 90% of the control loops are PID or triple term types. With its simplicity, clear functionality, applicability and ease of use, PID controllers remain the most popular approach for industrial process applications. However, they do not guarantee satisfactory performance against load disturbances and / or system parameter changes [2]. Mixed PID with adaptive and non-adaptive algorithms are used in electric motor control applications to consolidate and improve traditional fixed-gain PID controller performance [3]. Adaptive control is one of the widely used control strategies to design advanced control systems for better performance and accuracy [4]. Model reference adaptive control (MRAC) strategy is one of several techniques to solve control problems when controlled process parameters are little known or change during normal operation [5]. The MRAC technique was first proposed to solve the autopilot control problem [6]. The good results of the adaptive control method have increased the interest in the control method in recent years. On the other side, fuzzy Logic controllers are intuitive or based on experts' knowledge and experience. If basic information is not available, it may be difficult to create a fuzzy-based controller [5].

Sar and Dewan [6] designed the model reference adaptive PI controller for DC motor speed control and obtained the results by simulating the LabVIEW program. Hans Butler et al. [7] designed the model reference adaptive PI controller for direct-drive motor. The controller was tested on a direct-drive motor, and the results were compared with those obtained with a fixed proportional-integral-derivative (PID) controller. Platzer and Kaufman [8] applied to the speed control of a thyristor driven DC motor system, using model reference adaptive control. Yeniaydın et al. [9] designed the model reference adaptive PI controller for DC motor speed control and presented real time results. Barber et al. [10] designed adaptive controller for capable of interacting and controlling a physical system in real time. They used MATLAB and Simulink simulation environment for creating the models needed through block languages. Ali et al. [11] designed the adaptive PID controller for DC motor speed control and obtained the results in MATLAB-SIMULINK platform. Xh. Mehmeti [12] designed an adaptive PID controller for humanoid robot joints. The reason for using adaptive PID was that the humanoid robot had to track the trajectory with minimal error and no overshoot. Ge et al. [13] carried out constant temperature control of traditional dough making system with adaptive PID. They did a lot of experimental analysis to improve their self-tuning ability. As a result of these experimental analyzes, they concluded that the self-excited PID algorithm can be applied to the constant temperature system. Depending on the specific environment of the dough mixer, they presented experimental results of 3 mixes. Ghany et al. [14] introduced a new technique to adapt fractional order PID (FOPID) control based on the optimal model reference adaptive system (MRAS). The control technique they recommend was for power system load frequency control. This study proposed three methods for self-tuning FOPID control. The first method was implemented to tune the two integral and derivative parameters only and the rest of parameters were fixed. The second method was designed to adjust the proportional, integral derivative parameters while the other fractional parameters were constant. The last method was developed to adjust the five parameters of FOPID control simultaneously. Zhiwei et al. [15] aimed to control the FSM (fast steering mirror) by designing an adaptive controller. Due to the existence of flexible link in FSM, the uncertainty of the structural stiffness of the flexible-supported FSM was greatly increased when FSM worked. This led to a higher requirement for the robustness of control system. The numerical simulations results showed that the proposed algorithm can both suppress the influence of the change of structural stiffness and the sensor's noise. Rao et al. [16] applied PID and fuzzy logic methods for speed control of brushless DC motor. They summarized that PID controller was a simple controller with a simple tuning method but with a moderate response and performance. In addition, fuzzy controller was more complicated controller but with a good and more stable performance. Singh et al. [17] presented DC motor speed control based on boost converter with fuzzy logic control. They observed that the performance parameter of the system showed better transient and steady state performance using the fuzzy logic controller.

The aim and contribution of this study is designing and implementation of a fundamental adaptive control method for a basic control problem with a low cost microcontroller using rapid prototyping technic. In this study, MRAPIDC is designed by using MIT rule developed in 1960 by researchers from the Massachusetts Institute of Technology (MIT) for position control of a permanent magnet DC motor and applied with Matlab-Simulink supported Waijung block set and STM32F4 application development kit. Also, results of the PID and Fuzzy Logic control methods are compared with MRAPIDC.

In this paper, the DC motor model is briefly explained firstly. Then, the control methods are introduced in Chapter 3. Details of real time MRAPIDC implementation and experimental results of the all control methods are presented in Chapter 4. The results of the study are discussed in the last section.

2. Modelling of the System

DC motor is preferred for position control application because it has various size and power, cost effective and smaller. It is used frequently in robotic and industrial applications and it has easy driving technique compared to other motors.

Under the assumption of a homogeneous magnetic field, direct current (DC) motor has been modeled as a linear conversion from motor current to electrical torque. This classic model of the DC motor consists of a combined electrical and mechanical subsystem [5]. Thus, mathematical model is derived from the electrical and mechanical structure of the DC motor.

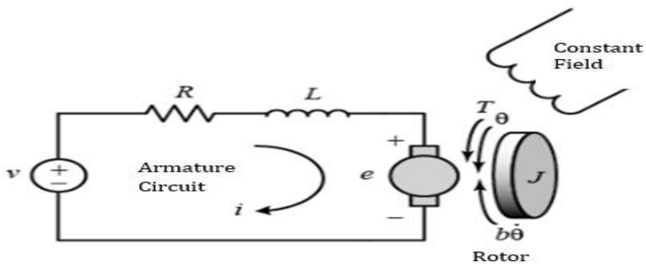


Figure 1. Electrical and mechanical structure of the DC motor [18].

The electric circuit of the armature and the free-body diagram of the rotor are shown in Figure 1. The electrical and mechanical equations of the DC motor are shown below in equations (1) - (4), respectively.

$$v(t) = Ri(t) + L\frac{di(t)}{dt} + e(t) \quad (1)$$

$$T(t) = J\frac{d^2\theta(t)}{dt^2} + b\frac{d\theta(t)}{dt} \quad (2)$$

$$T(t) = K_t\, i(t) \quad (3)$$

$$e(t) = K_e\, w(t) \quad (4)$$

where, *w (rad / sec)* and *i (A)* are respectively angular velocity and armature current. K_t (Nm / A), K_e *(V / rad / sec)* refer to torque and back-emf constants, which are dependent on the machine design. *B (Nms)* refers viscous damping. *T (Nm)* and *e (V)* represent torque and back-emf. *J (kgm²)* is the mechanical inertia included motor armature and shaft. *L (Henry)*, *R (ohm)* and *v (Volt)* respectively represent the inductance, the total connection resistance of the motor and the input voltage.

As can be seen, the generated equations have be written in *time (t)* domain. Laplace transforms of these equations have been taken and their forms in s domain have been written and system transfer function has been obtained from these equations.

The transfer function shows the relationship between the voltage applied to the motor, which is the input signal of the motor, and the output signal, which varies depending on the application. In the control applications in this article, the output signal is the rotor angle as the position of the motor has be controlled.

Transfer function giving the relationship between motor position and voltage:

$$\frac{\theta(s)}{V(s)} = \frac{K_t}{JLs^3 + (BL + JR)s^2 + (BR + K_tK_e)s} \quad (5)$$

Table 1 shows the parameters of the DC motor used in the experimental setup.

Table 1. Parameters of DC motor.

Parameter	Value	Unit
Rotor Resistance(R)	1.9	(ohm)
Rotor Inductance(L)	65	(μH)
Rotor Inertia Moment(J)	5.7	g.cm²
Motor Torque Constant(K_t)	13.4	mNm/A
Back-emf Constant	1.4	mV/rpm

3. Control Methods

3.1. PID Control

PID control is the most widely used control system among feedback control systems with its simplicity, clear functionality, applicability and ease of use. PID control is a control method that provides the system to fit to the desired reference value by processing the error obtained by subtracting the feedback signal from the reference value in parallel with proportional, integral and derivative effect [19]. P, I and D operations in the PID control system can be combined into different forms and become control methods such as P, PI, PD. The mathematical model of the PID control structure is given in equation (6).

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{d}{dt}e(t) \quad (6)$$

$$e(t) = r(t) - y(t) \quad (7)$$

where, *u(t)* indicates the control signal, *y(t)* indicates the output signal, *r(t)* indicates the reference signal and *e(t)* indicates the error signal. K_p, K_i and K_d coefficients in the equation are proportional, integral and derivative constants, respectively. If it is desired that the applied reference signal follow the output signal, the values of K_p, K_i and K_d must be determined correctly. In tuning the PID coefficients, they can be found using the closed loop Ziegler-Nichols method or the relay feedback method, or the PID controller can be designed by utilizing the root-locus curve as an acceptable performance by changing the parameters. In this study, PID parameters were obtained by closed loop Ziegler-Nichols method. With this method, the oscillation period (P_u) was 0.05 seconds and the K_{pMax} value that generates this oscillation was 8.3. Sampling time was selected as 0.01 seconds in the experiments.

Table 2. Determination of PID parameters according to closed loop Ziegler-Nichols method

	K_p	T_i	T_d
PID	0.6 K_{pMax}	0.5 P_u	0.125 P_u

$$K_i = \frac{K_p \boldsymbol{T}}{T_i} \qquad K_d = \frac{K_p\, T_d}{T} \quad (8)$$

When the values obtained were replaced in Table 2 and equation (8), K_p value was found to be 5, K_i value was found to be 2 and K_d value was found to be 3.125. When these calculated values were applied to the system, an overshoot and high rise time output was obtained. This is an expected result. Because the closed loop Ziegler-Nichols method does not guarantee no overshoot and low rise time [19]. These values were selected as initially and the parameters were changed to obtain the output to show the desired performance. The resulting parameter is: K_p value 5, K_i value 0.8 and K_d 0.1.

Figure 2 shows the PID control Simulink model designed with Waijung block sets and designed for the STM32F4 kit.

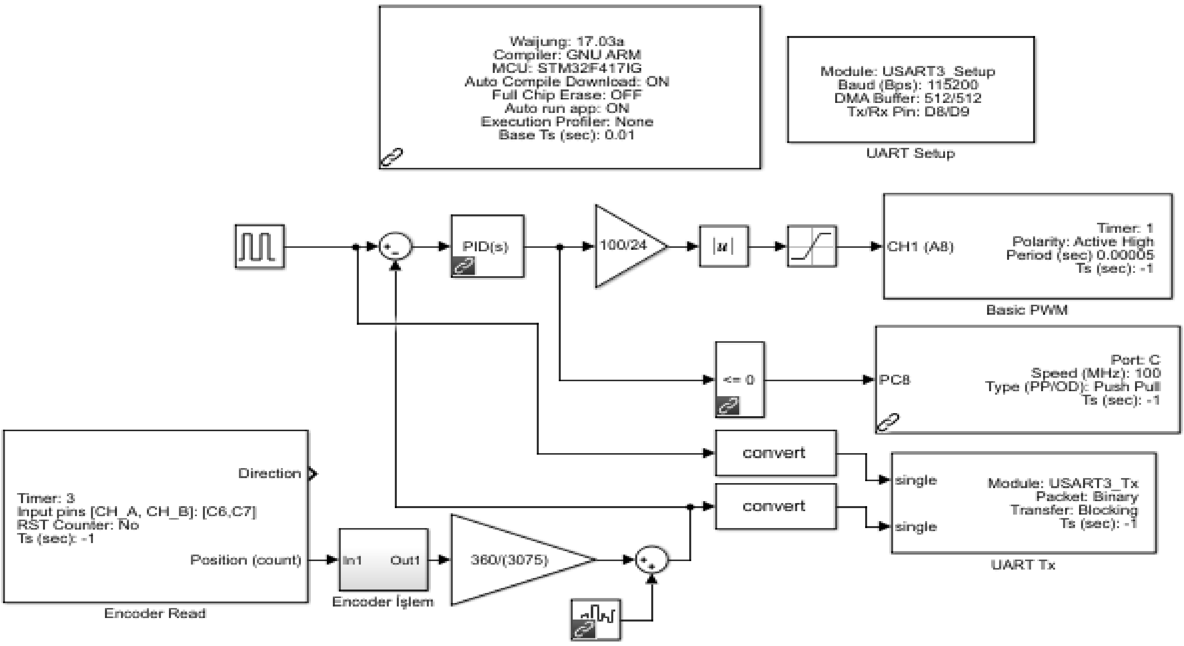


Figure 2. PID control Simulink model embedded in STM32F4 kit

3.2. Fuzzy Logic Control

Fuzzy logic, which is one of the intelligent control methods, is frequently used in practical applications. According to this theory, classical decision mechanisms have been replaced by fuzzy decision mechanisms. While classical logic is limited to absolute values such as 0 and 1, fuzzy logic takes into account values between 0 and 1. In other words, there may be infinite values representing the degree of membership between 0 and 1. This fuzzy logic decision structure gives us an output as a result of the conditions established by the experts familiar with the system and the results of these conditions. Especially in nonlinear systems, fuzzy logic is preferred with the structure that mimics the control structure of the human instead of the calculations with high processing load.

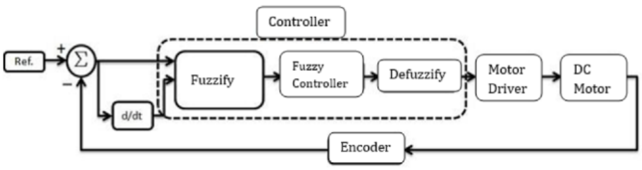
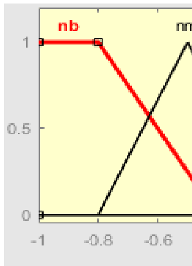
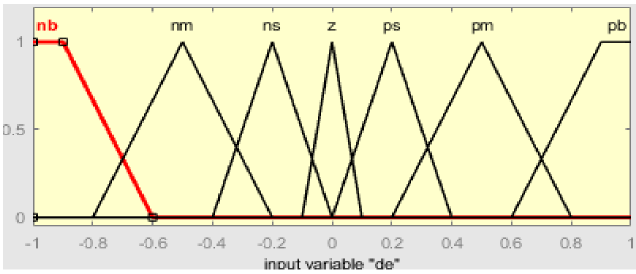


Figure 3. Basic fuzzy logic control block [20]

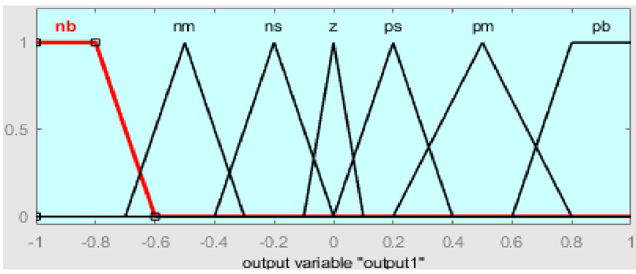
Figure 3 is a fuzzy logic control block diagram. As shown in Figure 3, the error is obtained by subtracting the instantaneous position values measured with the encoder from the reference in order to perform fuzzy logic control. The changing of error is obtained with the difference between the error and the previous value of the error. These two information are blurred by applying to membership functions. Then, inference is made according to the rules previously determined and the control signal is obtained by defuzzification.



(a)



(b)



(c)

Figure 4: (a) error (b) derivative of error (c) output membership functions

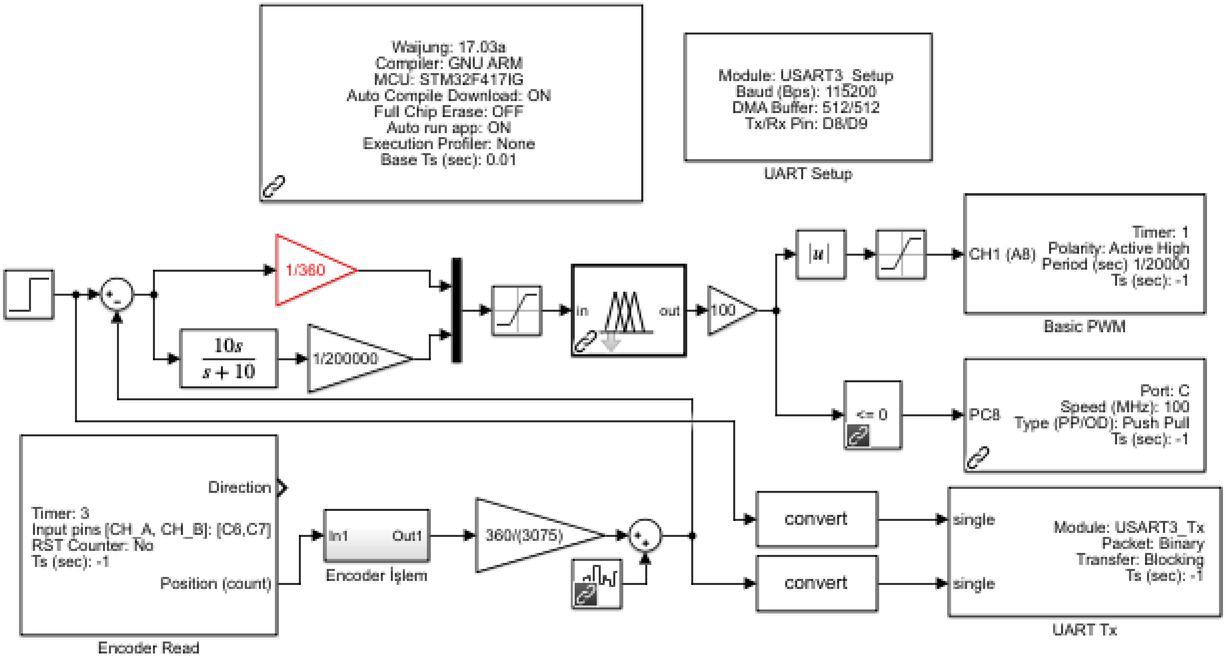


Figure 5: Fuzzy logic Simulink model embedded in the STM32F4 kit

Figure 4 shows the error, the change of the error, and the output membership functions generated by the MATLAB-FIS Editor.
As can be seen from Figure 4, the fuzzy membership functions are represented by seven different symbolic values. These are NB (Negative Big), NM (Negative Medium), NS (Negative Small), Z (Zero), PS (Positive Small), PM (Positive Medium) and PB (Positive Big).
Figure 5 shows the fuzzy logic Simulink model for the STM32F4 kit designed with Waijung block sets.
Figure 6 shows the fuzzy logic rule table.

e \ Δe	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NM	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NM	NM	NS	Z	PS	PM
Z	NB	NS	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PM	PB	PB	PB

Figure 6: Fuzzy logic rule table

3.3 Model Reference Adaptive Control

The model reference adaptive control (MRAC) was initially proposed to solve a problem in which features are given for a reference model that tells how the process output should respond to the command signal. The MRAC was proposed by Whitaker in 1958 as an important adaptive controller [5]. A block diagram of the MRAC is shown in Figure 7.
The desired output is generated by the input signal applied to the reference model that will give the desired output. Adjustment mechanism is designed to fit the system to the desired output. Even in the event of changes in system parameters or disturbing effects, the adjustment mechanism should guarantee that the system output is the same as the desired output [21].

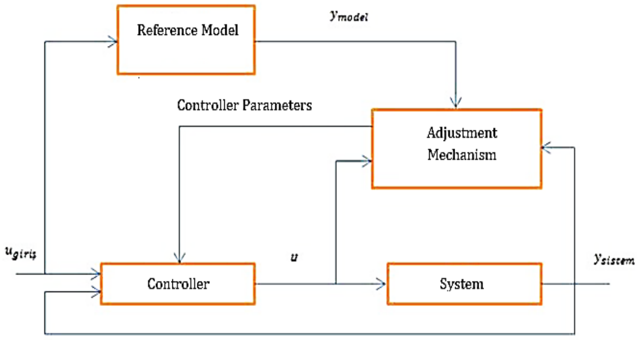


Figure 7: MRAC block diagram [9]

The controller presented in Figure 7 can be considered to consist of two loops. The ordinary feedback loop, called the inner loop, consists of a process and a controller. The second cycle, called the outer loop, consists of the adjustment mechanism that changes the controller parameters and the reference model.
Determining the adjustment mechanism to obtain a stable system to make the error zero is an important problem with the MRAC system [5]. In this article, MIT rule in equation (11) was used for parameter adjustment mechanism.

$$e = y_{system} - y_{model} \tag{9}$$

e is a tracking error. The cost function, which is the function of the parameter vector (θ), is shown below.

$$J(\theta) = \frac{1}{2} e^2(\theta) \tag{10}$$

The reduction of the cost function means that the tracking error (e) decreases and the system output approaches the desired output. Therefore, an equation that will minimize the cost function is needed. This equation [21] is shown below in equation 11. The constant γ in the equation is known as adaptation gain.

$$\frac{d\theta}{dt} = -\gamma\frac{\partial J}{\partial \theta} = -\gamma\frac{\partial J}{\partial e}\frac{\partial e}{\partial \theta} = -\gamma e\frac{\partial e}{\partial \theta} \tag{11}$$

3.3.1 MRAPID Design

The order of the position-dependent transfer function of the motor can be considered as second order because motor inductance is very low in order to simplify controller design [6], [22]. a_0, a_1, a_2 and b_0 are constants in the equation.

$$G(s) = \frac{y_s\,(s)}{u\,(s)} = \frac{b_0}{a_2 s^2 + a_1 s^2 + a_0} \tag{12}$$

In the frequency domain, the PID transfer function is given in equation (14).

$$e_p = u_{input} - y_{system} \tag{13}$$

$$\frac{u\,(s)}{e_p\,(s)} = \frac{K_d s^2 + K_p s + K_i}{s} \tag{14}$$

The closed loop transfer function is as follows.

$$\frac{y_s\,(s)}{u_g(s)} = \frac{b_0(K_d s^2 + K_p s + K_i)}{a_2 s^3 + (a_1 + b_0 K_d)s^2 + (a_0 + b_0 K_p)s + K_i} \tag{15}$$

The MIT rule is finally written as follows.

$$\frac{d\theta}{dt} = -\gamma e\frac{\partial e}{\partial \theta} = -\gamma e\frac{\partial e}{\partial y_s}\frac{\partial y_s}{\partial \theta} = -\gamma e\frac{\partial y_s}{\partial \theta} \tag{16}$$

$$\theta = \left(-\frac{\gamma}{s}\right) e\,\frac{\partial y_s}{\partial \theta} \qquad\qquad \theta = K_p, K_i, K_d \tag{17}$$

$\frac{\partial y_s}{\partial K_p}$ in equation (17) is obtained by the product of the inner and the outer of equation (15). The application of the MIT rule to the K_p control parameter was obtained as below.

$$Y_s\,(s)\left[a_2 s^3 + (a_1 + b_0 K_d)s^2 + (a_0 + b_0 K_p)s + K_i\right] = U_g\,(s)\left[b_0(K_d s^2 + K_p s + K_i)\right] \tag{18}$$

By taking partial derivative of equation (18) according to K_p, K_i and K_d, equations (19), (20) and (21) were obtained.

$$\frac{\partial y_s}{\partial K_p} = \frac{b_0 s(u_g - y_s)}{s^3 + \left(\frac{a_1 + b_0 K_d}{a_2}\right)s^2 + \left(\frac{a_0 + b_0 K_p}{a_2}\right)s + \frac{K_i}{a_2}} \tag{19}$$

$$\frac{\partial y_s}{\partial K_i} = \frac{b_0(u_g - y_s)}{s^3 + \left(\frac{a_1 + b_0 K_d}{a_2}\right)s^2 + \left(\frac{a_0 + b_0 K_p}{a_2}\right)s + \frac{K_i}{a_2}} \tag{20}$$

$$\frac{\partial y_s}{\partial K_d} = \frac{b_0 s^2(u_g - y_s)}{s^3 + \left(\frac{a_1 + b_0 K_d}{a_2}\right)s^2 + \left(\frac{a_0 + b_0 K_p}{a_2}\right)s + \frac{K_i}{a_2}} \tag{21}$$

When equations (19), (20) and (21) are re-arranged with equation (17), equations (22), (23) and (24) are obtained.

$$K_p = \left(-\frac{\gamma_p}{s}\right) e\,\frac{b_0 s(u_g - y_s)}{s^3 + \left(\frac{a_1 + b_0 K_d}{a_2}\right)s^2 + \left(\frac{a_0 + b_0 K_p}{a_2}\right)s + \frac{K_i}{a_2}} \tag{22}$$

$$K_i = \left(-\frac{\gamma_i}{s}\right) e\,\frac{b_0(u_g - y_s)}{s^3 + \left(\frac{a_1 + b_0 K_d}{a_2}\right)s^2 + \left(\frac{a_0 + b_0 K_p}{a_2}\right)s + \frac{K_i}{a_2}} \tag{23}$$

$$K_d = \left(-\frac{\gamma_d}{s}\right) e\,\frac{b_0 s(u_g - y_s)}{s^3 + \left(\frac{a_1 + b_0 K_d}{a_2}\right)s^2 + \left(\frac{a_0 + b_0 K_p}{a_2}\right)s + \frac{K_i}{a_2}} \tag{24}$$

$$\gamma_p = -0.15, \qquad \gamma_i = -0.07\,, \qquad \gamma_d = -0.01$$

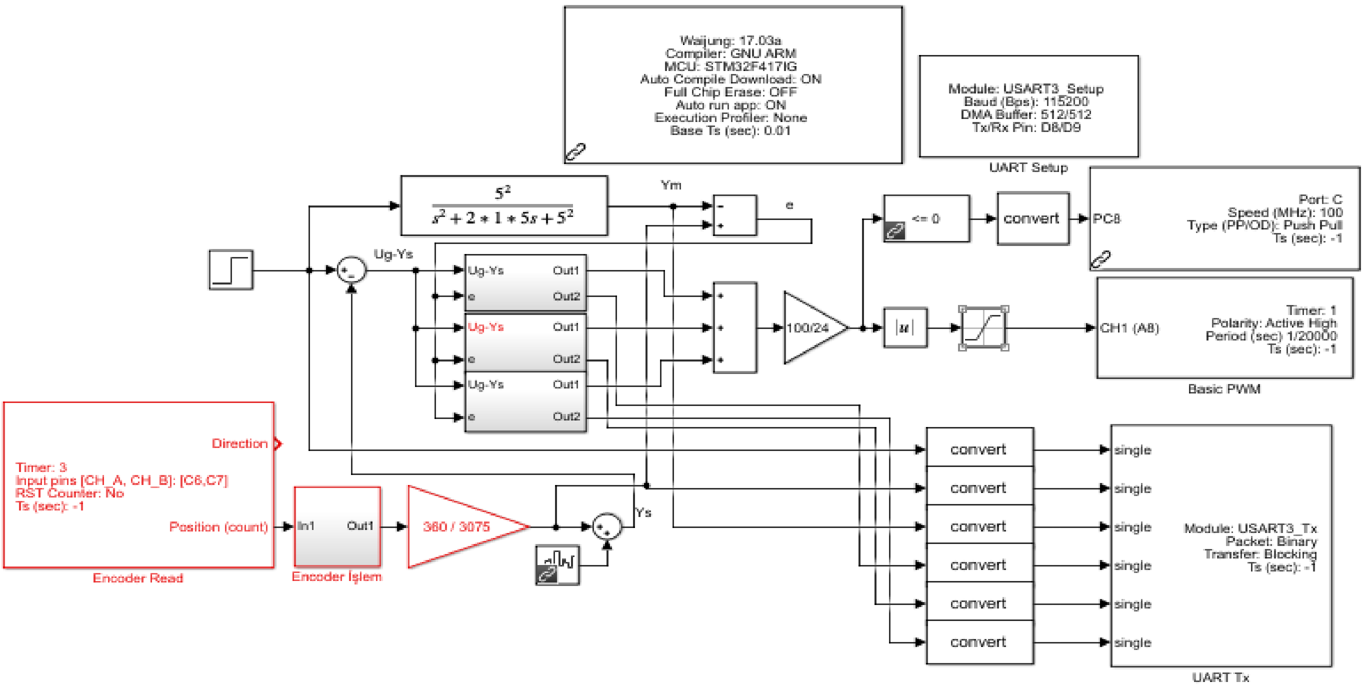


Figure 8. MRUPID Simulink model embedded in the STM32F4 kit

4. Real Time MRAPID Implementation

Figure 8 shows the MRAPID Simulink model designed with the Waijung block sets for the STM32F4 kit. Details of the controller parameter adjustment mechanisms are shown in figure 9, 10 and 11.

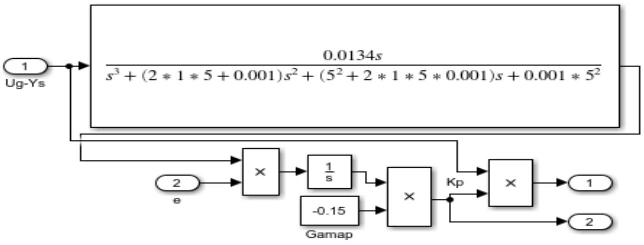


Figure 9. K_p adjustment mechanism sub-model

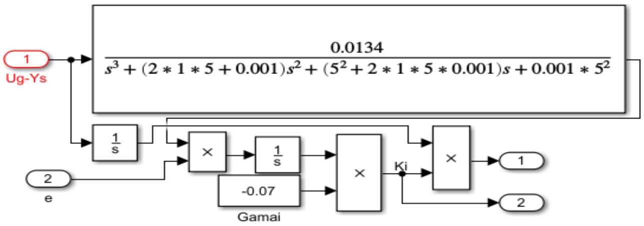


Figure 10. K_i adjustment mechanism sub-model

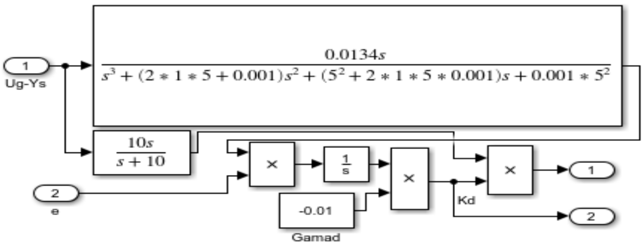


Figure 11. K_d adjustment mechanism sub-model

Since the poles of the closed-loop transfer function affect the behavior of the system response, the roots of the characteristic equation are important. The displacement of the poles determines the characteristics of the system such as settling time, rise time, and overshoot. The transfer function of a second order system with the desired characteristic can be formed by ζ (damping ratio) and w_n (natural frequency). Using this feature, the transfer function of the reference model is selected as the second order, which is given in equation (25). Since overshoot and oscillation are not desired, ζ and w_n value was selected as 1 and 5 rad/s, respectively.

$$G_m = \frac{w_n}{s^2 + 2\zeta w_n + w_n^2} \quad (25)$$

Given that the process parameters a_0, a_1, a_2, and b_0 are unknown, formulas derived using MIT rule (11) cannot be used. Instead, some approaches are required [5]. Considering that the system will follow an excellent model by comparing the input-output relationships of the system with the reference model, the following approaches will be used [5].

$$s^3 + \left(\frac{a_1 + b_0 K_d}{a_2}\right) s^2 + \left(\frac{a_0 + b_0 K_p}{a_2}\right) s + \frac{K_i}{a_2} \equiv (s + \sigma)(s^2 + 2\zeta w_n + w_n^2) \quad (26)$$

Since the closed-loop transfer function in equation (15) has a higher degree than the reference model (25), it should be known that an arbitrary selection for the location of the stable pole (-σ) greatly influences the performance of the algorithm and was chosen to be close to the origin as 0.001. The result transfer functions are as follows.

$$G_i(s) = \frac{b_0}{s^3 + (2\zeta w_n + \sigma)s^2 + (w_n^2 + 2\zeta w_n \sigma)s + \sigma w_n^2} \quad (27)$$

where $G_p(s) = G_d(s) = sG_i(s)$ and $b_0 = K_t$.

4.1 Experimental Results

Figure 12 shows the experimental setup in which the applications are tested. In the experimental setup, a permanent magnet Faulhaber brush DC motor with 12V supply with a maximum speed of 120 RPM and a ratio of 1:66 gearbox was used. In addition, a two-channel optical encoder with a resolution of 12 CPR (Counts per Revolution) is used in the motor. The setup also includes the STM32F4 MCU development kit, the LMD18200 motor driver with H-bridge structure and the USB-UART converter for serial communication. The reasons for choosing STM32F4 application development kit is its low cost, having 32 bit ARM processor, programming and debugging units on the kit, being easy to learn and being programmable quickly with blocks in the code-free Waijung block set via Simulink.

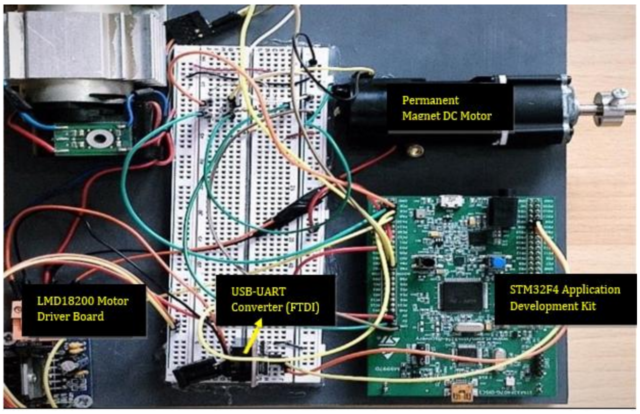


Figure 12. Experimental setup

While real-time responses of the PID and Fuzzy logic controllers under the step function reference were obtained, real-time responses of the MRAPID controller were obtained under both the step function and the sine wave reference inputs. Also, in MRAPID, variations of Kp, Ki and Kd parameters in time were obtained and given.
The step function inputs are a reference with a delay of 1 second and amplitude of 360°, while the sine wave is a

reference with a period of 4 seconds ranging from 0° to 360°. In addition, noise signal with 0.1 noise power and a sampling time of 0.1 seconds was introduced to the encoder outputs.

The responses obtained under the step function reference input are shown in figure 13 and 14.

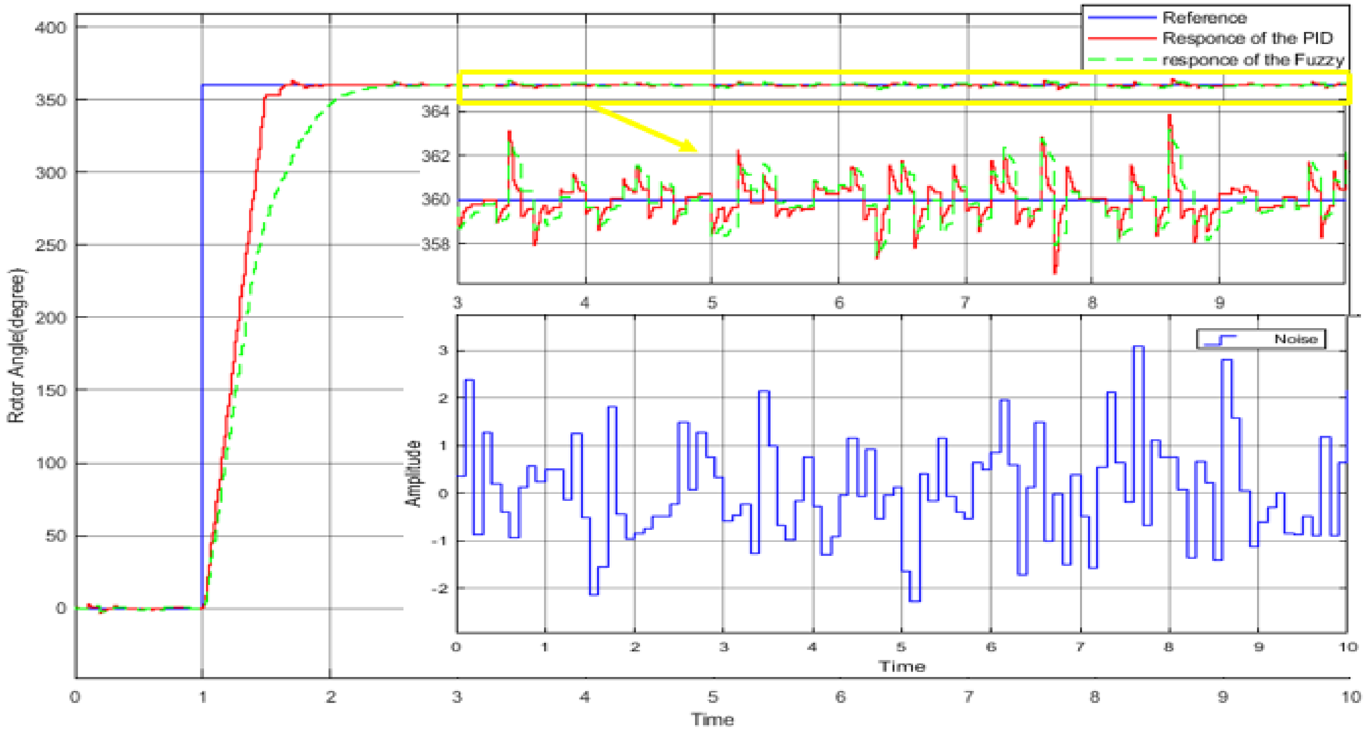


Figure 13. Step function response of the PID and fuzzy logic controller with noise signal added to encoder outputs

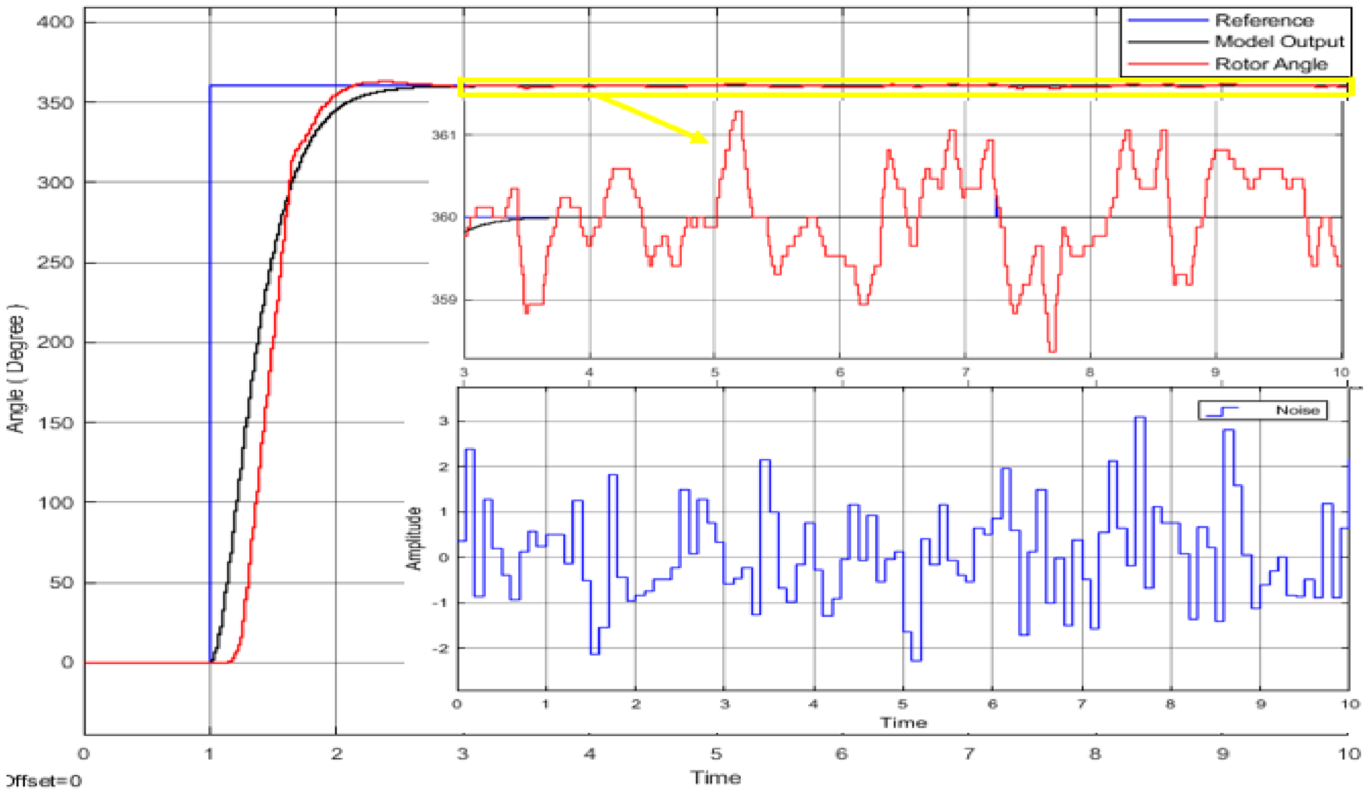


Figure 14. Step function response of MRAC controller with noise signal added to encoder outputs

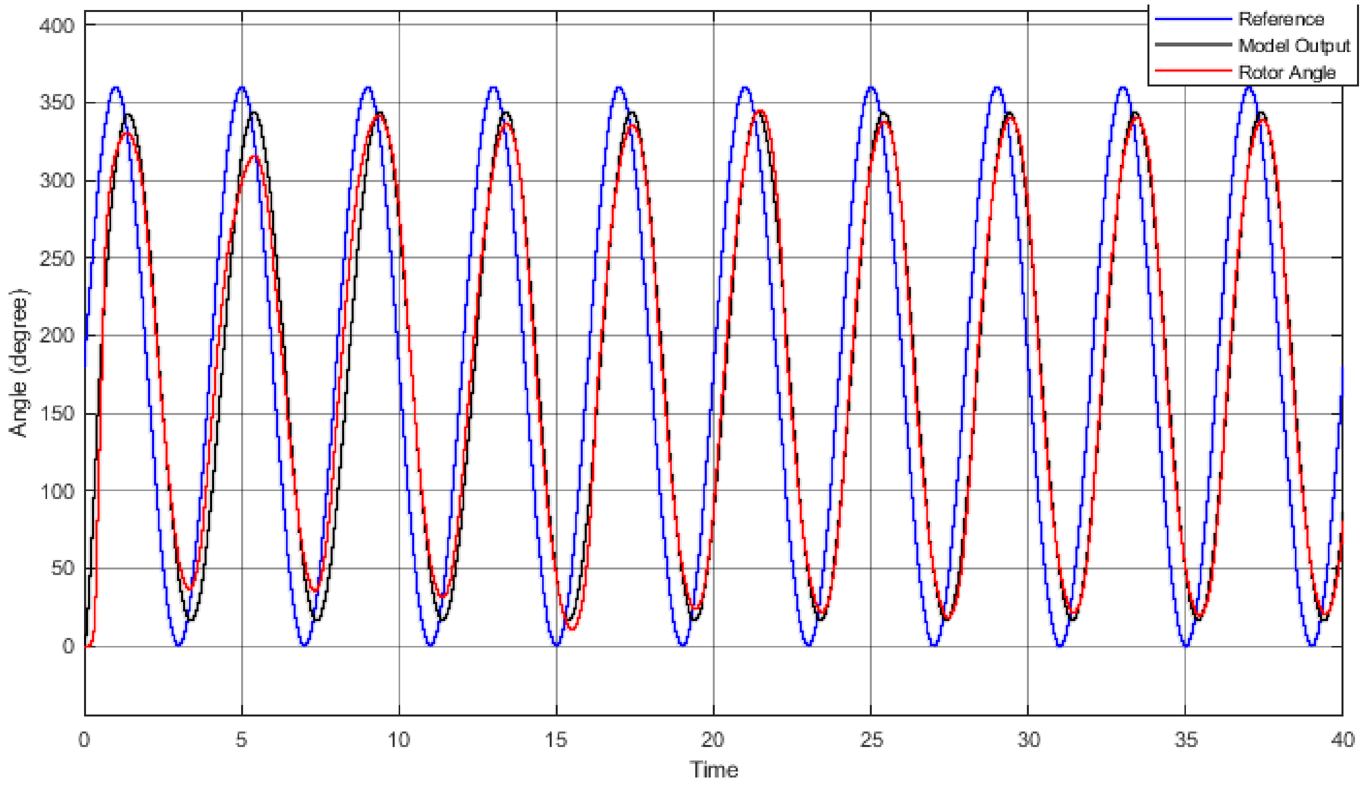


Figure 15. Sine function response of MRAC controller

Table 3. Comparison table of MSE, RMSE and MAE

	Mean Square Error (MSE)	Root Mean Square Error (RMSE)	Mean Absolute Error (MAE)
Fuzzy	2923	54,07	14,20
PID	2299	47,95	9,92
MRAPID	310,20	17,61	4,85

Table 3 shows the MSE, RMSE and MAE values of the Fuzzy, PID and MRAPID controllers. As can be seen, all error values of MRAPID were calculated lower than Fuzzy and PID.

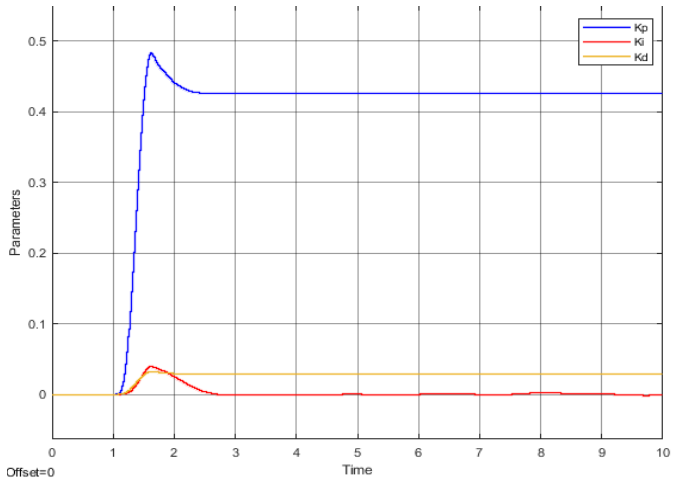


Figure 16. Change of K_p, K_i and K_d values of MRAC controller's step function response

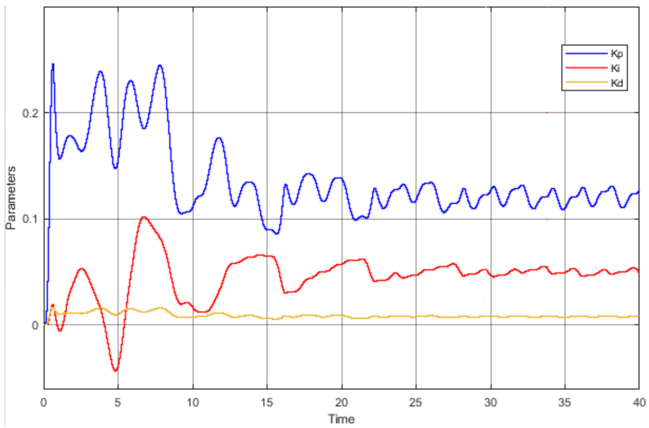


Figure 17. Change of K_p, K_i and K_d values of MRAC controller's the sine function response

Figure 16 shows the change of Kp, Ki and Kd values of the MRAPID controller against the step input. As can be seen, about 1.5 seconds after the reference signal is given to the system, the parameters are set to approximately constant values.

The output of the MRAPID controller corresponding to the sine reference input is shown in Figure 15 and the change of K_p, K_i and K_d values is shown in Figure 17.

5. Discussion

The desired output is generated by the input signal applied to the reference model that will give the desired output. Determining the adjustment mechanism to obtain a stable system to make the error zero is a major challenge for the MRAC system. Therefore, the adjustment mechanism is designed to fit the system to the desired output. The system output reaches the desired output with a very small error after the transient response formed at the beginning of the system due to the difference between the desired output and the system output. Even if there is a change in system parameters or disturbing effects, such as adding noise to the encoder output, the adjustment mechanism is working to match the system output to the desired output.

In addition, the results obtained from PID and Fuzzy logic controllers are given. Performances are observed by adding noise signals with 0.1 noise power and 0.1 second sampling time to the encoder outputs. In order to make comparison with PID and fuzzy logic controllers, MRAPIDC, PID and Fuzzy logic control systems have a step input with a 360° amplitude. Results showed that the MRAPID controller was less affected by noise and tried to follow the reference model. K_p, K_i and K_d values were adopted so that the output will able to follow the model. Approximately 1.5 seconds after the reference signal was applied, the parameters were settled to approximately constant values. At the same time, a sine wave input with a 360° amplitude and a frequency of 0.25 Hz was applied to examine transient and steady state behaviors. System output followed the model with a varying error in the transient regime, then the error gradually decreased after approximately 8 seconds and the error decreased to an acceptable level after approximately 20 seconds.

6. Conclusion

In this study, PID, Fuzzy logic and MRAPID controllers are designed and applied in real time for position control of a DC motor. Experimental results obtained in this study using a brushed DC motor showed that MRAPID controller forces the system to follow the reference against the noise added to the system, which may represent changes in environmental conditions or disruptive effects. MRAPID controller was less affected by measurement noise compared to others. MRAPID can also be used as an auto-tuner for similar systems. In future studies, Lyapunov function based robust adaptive PID controller can be designed.

7. References

[1] Bingül, Z. and Küçük, S., *Robot Dinamiği ve Kontrolü*, Umuttepe Yayınları, Kocaeli, 2017.

[2] Xiong, A. and Fan, Y., "Application of a PID Controller using MRAC Techniques for Control of the DC Electromotor Drive", *International Conference on Mechatronics and Automation*, 2007, pp. 2616-2621.

[3] Elsodany, N. M., Rezeka, S. F., and Maharem, N.A., "Adaptive PID control of a stepper motor driving a flexible rotor", *Alexandria Engineering Journal*, Vol. 50, pp. 127-136, 2011.

[4] Jain, P., and Nigam, D. M. J., "Design of a Model Reference Adaptive Controller Using Modified MIT Rule for a Second Order System", *Advance in Electronic and Electric Engineering*, Vol. 3, pp. 477-484, 2013.

[5] Nikranjbar, A., "Model Reference Adaptive PID Control of Servo Speed DC Motor", *Majlesi Journal of Mechatronic Systems,* Vol. 2, pp. 7-13, 2013.

[6] Sar, S. K., Tech, M., and Dewan, L., "MRAC Based PI Controller for Speed Control of D.C. Motor Using Lab View", *WSEAS TRANSACTIONS on SYSTEMS and CONTROL*, Vol. 9, pp. 10-15, 2014.

[7] Butler, H., Honderd, G., and van Amerongen, J., "Model reference adaptive control of a direct-drive DC motor", *IEEE Control Systems Magazine*, Vol. 9, pp. 80-84, 1989.

[8] Platzer, D. and Kaufman, H., "Model Reference Adaptive Control of Thyristor Driven DC Motor Systems Subject to Current Limit a Tions", *IFAC Proceedings Volumes*, Vol. 17, pp. 1991-1995, 1984.

[9] Yeniaydın, Y., Sakacı, B., Yaren, T., Süel, V., and Kizir, S., "DC Motor Hız Kontrolü için Model Referans Uyarlamalı PID Denetleyici Tasarımı", *Türk Otomotiv Konferansı*, 2014, pp. 313-319.

[10] Barber, R., Rosa, D. R. and Garrido, S., "Adaptive control of a DC motor for educational practices", *IFAC Proceedings Volumes, Vol.* 46, pp. 244-249, 2013.

[11] Ali A. T. and Tayeb, E. B. M., "Adaptive PID Controller for Dc Motor Speed Control", *International Journal of Engineering Inventions,* Vol. 1, pp. 26-30, 2012.

[12] Mehmeti, Xh., "Adaptive PID controller design for joints of Humanoid Robot", *IFAC-Pap.*, Vol. 52, pp. 110-112, 2019.

[13] Ge, L., Liu, B., and Wang, T., "Improvement of surface temperature control system based on fuzzy adaptive PID algorithm", *International Conference on Robotics, Intelligent Control and Artificial Intelligence - RICAI,* 2019, pp. 368-373.

[14] Shamseldin, M. A., Sallam, M., Bassiuny, A. H., and Ghany, A. M. A., "A novel self-tuning fractional order PID control based on optimal model reference adaptive system", *International Journal of Power Electronics and Drive System (IJPEDS)*, Vol. 10, pp. 230-241, 2019.

[15] Zhiwei, A., Jianbo, J., Pengju, W., Ruijing, L., and Hua, Z., "Design of model reference adaptive control for fast steering mirror based on generalized error fast differential method", *5th International Conference on Advanced Computing, Networking and Security (ADCONS)*, 2019, pp. 18-26.

[16] Rao, N. A., and Kumar, D. C. R., "Speed Control of Brushless Dc Motor by Using PID and Fuzzy Logic Controller", IJIRT, Vol. 6, pp. 72-77, 2019.

[17] Singh, A. kumar, Saxena, A., Terang, P. Poon, Tulsyan, P., and Waris, M., "Speed Control of DC Motor Using Chopper Based on Fuzzy Logic", IOP Conf. Ser. Mater. Sci. Eng., 2019, pp. 1-8.

[18] "Control Tutorials for MATLAB and Simulink - Motor Position: Simulink Modeling". [Çevrimiçi]. Erişim adresi: http://ctms.engin.umich.edu/CTMS/index.php?example=MotorPosition&section=SimulinkModeling. [Erişim: 21-Kas-2019].

[19] Kizir, S., Yaren, T. and Kelekçi, E., *Gerçek Zamanlı Kontrol*, Seçkin Yayınevi, Ankara, 2019.
[20] Köse, F., Kaplan, K. and Ertunç, H. M., "PID ve Bulanık Mantık ile DC Motorun Gerçek Zamanda STM32F407 Tabanlı Hız Kontrolü", *Türk Otomotiv Konferansı*, 2013, pp. 1178-1183.
[21] Åström K. J. and Wittenmark, B., *Adaptive Control: Second Edition*, Courier Corporation, 2013.
[22] Machine Theory, System Dynamics and Control Division, "Laboratory Manual - Position Control of Rotary Servo Base Unit using PIV Controller", Department of Mechanical Engineering, Yildiz Technical University.