Hexapod Robot Design and Performance Comparison of Fuzzy and PID Control Methods

S. AYDIN, L. GOKREM and M. S. CAN

***Abstract*— In this study, a six-leg spider robot (hexapod) was designed and controlled for greenhouse, search and rescue operations and military applications. Solidworks design program was used in the design stage and Matlab Simulink program was used in the control stage of the robot. In the study, a motion trajectory was determined by interpolation technique for the robot and it was aimed to move on this trajectory. Proportional-Integral-Differential (PID) controller and Fuzzy Logic Controller (FLC) were used for the trajectory control of the robot. The robot walk movements were applied in both control types with motion on flat ground, motion on inclined ground, motion at different weights and different friction coefficients. In the analysis studies, the total weight of the robot was taken into consideration as 4kg. As a result of the analysis studies, it was observed that this robot, which was designed and analyzed, followed a trajectory defined by a mass of 4 kg with an error value of 1 mm on average. During the trajectory tracking, it was found that the Fuzzy controller performs better than the PID controller at the turning points of the reference trajectory curve of the robot.**

Index Terms— **Solidworks, Matlab, PID, fuzzy logic, Simulink, hexapod robot.**

I. Introduction

THE WORD robot, which has entered many areas of our lives, was first used in history by Czech writer Karel Capek in a playwriting in 1921. In the Czech language, the word robot means captive, slave. Nowadays, the word robot is used for machines that can behave like human beings and can be used as an alternative to human beings.

Throughout history, human beings have endeavored to develop tools to help them, and tried to make animals and plants like devices, especially similar to human. Robot experiments were created with mechanical devices in different parts of history and the actual development emerged in the 20th century. In the 1900s, robots came into our lives as auxiliary work machines in weaving looms [1].

With the development of technology, the studies on autonomous control of the systems have accelerated and thus the power of production systems is changing and developing day by day. The most important factors of this development and change are undoubtedly the robots used in the systems. Autonomously controlled robots are capable of processing and receiving images without any external effects. Since they are stable and can work continuously, the quality of the results is minimized by reducing the margin of error. Robots are widely used in medicine, defense, aircraft and space studies outside the industry.

Hexapod robots include legs attached to six robot bodies. The legs are controlled to certain degree, so that the robot can move within the environment and perform the specified tasks. Such robots are suitable for terrestrial and space studies. They include versatile movement, variable geometry, good stability, and durable movement in various terrains [2,3]. Hexapod robots are the subject of many applications because they can overcome larger obstacles than wheeled robots of the same size. The use of wheels or pallets limits movement if there are obstacles that can extend up to half the diameter of the wheels. In contrast, legged robots can overcome comparable obstacles as the foot length is longer than a wheel diameter [2,4].

Nowadays, in many machine production processes, a model of the machine is made with 3D computer aided design (CAD) program and also heating, abrasion and strength tests are performed on this model with analysis programs. As a result of the analysis, the most suitable design model is obtained and with this method, rapid and less costly prototypes can be obtained. In this study, the 3D design of a six-legged hexapod (spider like) robot and its control on Matlab / Simulink base were used to control the factors such as spraying, irrigation or collecting at the right time in the greenhouse farming area without any external intervention, autonomous control, and transferring images from areas where human health might be at risk. For the movement of the robot, the points that the robot must go within the motion range of the robot are defined and a trajectory is defined for these points by using curve fitting method. FLC and PID controllers were used to monitor the trajectory of the robot and the control performances of

SEFA AYDIN, is with Department of Mechatronic Engineering, Tokat Gaziosmanpasa University,Tokat, TURKEY
(e-mail: sefa.aydin.29@hotmail.com).

LEVENT GÖKREM, is with Department of Mechatronic Engineering, Tokat Gaziosmanpasa University,Tokat, TURKEY
(e-mail: levent.gokrem@gop.edu.tr).
https://orcid.org/0000-0003-2101-5378

MEHMET SERHAT CAN, is with Department of Electrical-Electronic Engineering, Tokat Gaziosmanpasa University,Tokat, TURKEY
(e-mail: mehmetserhat.can@gop.edu.tr).
https://orcid.org/0000-0003-2356-9921

Manuscript received November 25, 2019; accepted December 12, 2019.
DOI: 10.17694/bajece.650784

these two controllers were compared. The actual prototype of the robot must be equipped with cameras for observation and application-specific sensors for data collection. In this way, images taken from the camera can be used in processes such as obstacle recognition, analysis of plant leaves, mine detection and image transfer using image processing and machine learning algorithms.

II. MATERIAL

A. Simulation Softwares

SolidWorks is a three-dimensional design program that uses Microsoft Windows graphic interfaces. SolidWorks is now widely used in mechanical designs. The design phase consists of a special opened part command for each part and assembly commands that combine these parts. It begins with a two-dimensional drawing at the beginning of a design. After the main body of the part is drawn, three dimensionality is given to the drawing by means of commands. Iron and steel materials can be assigned to these designed parts. Parts can be painted in desired colors. Fittings are combined in the "assembly" section of the completed design. The relationship to the point of contact of each part is defined. These relations are concentric, tangent to each other or coincident. It is possible to make corrections and new part designs on the parts when the assembly stage is started [4].

Matlab mathematical programming language is developed by MathWorks. Matlab, which contains many ready-made commands, allows users to work with programs created in different languages such as C, C ++ and Java. Matlab is now widely used in engineering.

B. PID Controller

Control with PID controller is a common control method in robotics and automation systems. This method receives continuous feedback from the system and calculates the error value between the reference point and the controlled value within the specified time and generates output values to reduce the error to zero. Fig. 1 shows a closed-loop PID controller control block diagram. In general, in a closed-loop control structure, the system output signal $y(t)$ is intended to be output to a reference value $r(t)$ in the shortest possible time, with minimal overrun. In addition, $y(t)$ should be able to hold in spite of external disturbances at this reference value. This is called durability. The PID controller takes the error value $e(t)$ as an input and finds the control signal $u(t)$ by summing the results obtained using proportional (P), integral (I) and derivative (D) operators. Finally, the control sign $u(t)$ is applied to the system. Thus, the error value $e(t)$ is intended to be minimized.

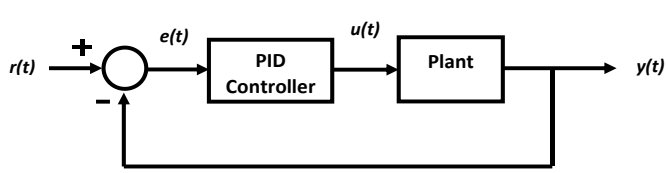


Figure 1. PID controller basic diagram.

The PID controller's performance is determined by the proportions, derivative and integral operators' gain coefficients. These gain coefficients are expressed as K_p, K_i and K_d, respectively. These gain values very important for rise time, overshoot rate, peak value, settling time and steady state error values of the system. Some energy storage systems do not respond immediately to input signals and react temporarily over time [5]. Therefore it is very important to tuning the gain coefficients of the PID controller. The expressions of the proportional, integral and derivative operators in the time domain are as shown in Equ.(1) -Equ.(4).

$$P_{operator} = K_p e(t) \tag{1}$$

$$I_{operator} = K_i \int_{t1}^{t2} e(t)\,dt \tag{2}$$

$$D_{operator} = K_d \frac{de(t)}{dt} \tag{3}$$

$$u(t) = K_p e(t) + K_i \int_{t1}^{t2} e(t)\,dt + K_d \frac{de(t)}{dt} \tag{4}$$

The expression of the control signal with PID controller in the Laplace space is given in Equ. (5) below and the transfer function of the PID controller in the Laplace space is given in Equ. (6).

$$U(s) = \left(K_p + \frac{K_i}{S} + K_D s\right) E(s) \tag{5}$$

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{S} + K_D s \tag{6}$$

The degrees of the I and D operators given above are expressed as integers and the representation is called integer order PID or PID. The degrees of the I and D operators are not integer degrees, λ for the integral operator and μ for the derivative operator. These two parameters are in the range of [0,1] and in this case the fractional order PID controller is obtained. In the fractional order PID design, in addition to the P, I and D parameters, two additional parameters indicated by λ and μ are added, and this allowing the use of five parameters. Fractional integral and derivative operations are found in L'Hospital's work [6,7].

C. Fuzzy Logic Controller

FLCs can be said to be one of the most popular methods used to control nonlinear systems in particular. FLC is one of the model independent methods. Using the expert knowledge in control design for the strengths of FLCs, it can be said that the irregularity system can increase both its flexibility and robustness in the presence of irregularities and uncertainties. Because of these properties, FLCs are preferred in the control problems of mobile robots, especially in the control of unmanned aircraft, which, as a more challenging control

problem, previously forced the acquisition of mathematical model parameters [8-14].

In FLCs, input variables are correlated with each other on the basis of fuzzy set theory, resulting in output variables. The way in which input variables are associated and what results are obtained from these relationships are performed by rules. Rules are created within the framework of the user's knowledge and expertise. An FLC creates four basic units; fuzzification, rules, inference, defuzzification units. An FLC architecture is shown in the diagram in Fig. 2.

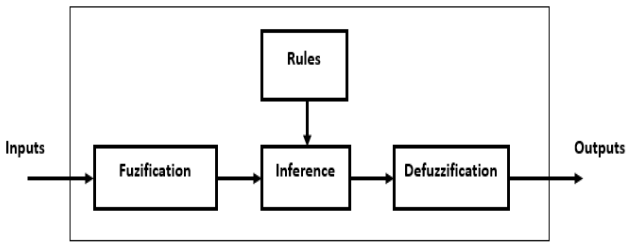


Figure 2. FLC system architecture.

Fig. 2 shows a FLC structure. In FLC, the input data is first converted to fuzzy subsets within the [0.1] range of the blur step. Membership functions are used for this conversion. First, several fuzzy subsets are defined for each input variable, including the range of change of the input variable. The instantaneous value of the input variable, whichever fuzzy subset corresponds, applies to the membership function of that subset and fuzzy values are calculated.

Another step in FLC design is the rule base. Rules are created according to the expertise of the designer. These rules are in the form of "*if-then*" definitions. In the inference phase, fuzzy outputs are obtained according to fuzzy input values and rules. The final step in FLC design is the rinsing step. In the rinsing step, the process of converting the fuzzy results obtained by the extraction step into a real and controllable system is performed. The most well-known methods are "*Mamdani*", "*Sugeno*" and "*Tsukamoto*".

The designed FLC is used in place of the PID controller in the closed loop control structure given in Fig. 1. The FLC controller produces more effective results than the PID controller, especially in the control of systems involving parameter changes.

III. METHOD

Solidworks design program was used in the robot design stage and Matlab R2017b / Simulink libraries were used in the motion control stage of the robot. The design is based on the spider anatomical structure and the leg structures are designed to be 60 degrees apart to maximize the balance factor. In the design, the telescopic camera carrier, which can be extended and shortened, is placed in the center of gravity of the robot for different applications. The robot is intended to move on an orbit. For this reason, an orbit is defined by curve fitting method. Thus, the robot can be sent to the desired coordinates in the *x-y* plane in the cartesian coordinate system in the desired time. Since the robot moves in two dimensions in the *x-y* plane, a controller for trajectory control on the *x*-axis and a second controller for control on the *y*-axis are used. In the *x*-axis, a linear signal is referred to as the reference, and the *y*-axis is referred to as a path (trajectory) of determined points. The autonomous motion control of the robot on these trajectories has been done with PID and FLC controllers separately in Simulink environment. The tripod gait was used for the stepping motion of the robot. Wave stepping is a form of walking motion that is commonly used in spider robots. Fig. 3 shows tripod gait movements.

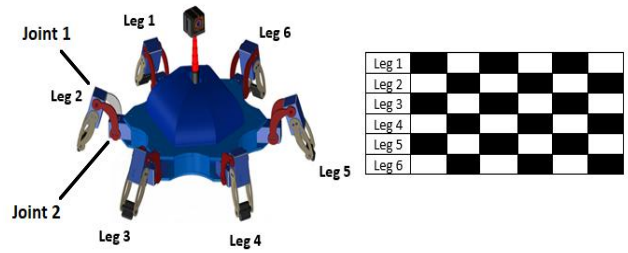


Figure 3. Hexapod robot and tripod gait motion.

As shown in Fig. 3, each leg of the robot has two joints. There are servo motors in these joints to provide movement. All of these servo motors are centered at an angle of 60 degrees during assembly. The first joints in the legs connect the foot to the body of the robot and at the same time move it back and forth in the *x-y* plane. The second joint performs the up and down movement of the foot in the *x-z* plane. In the basic robot movement, according to the table order in Fig. 3, firstly, two joints in the 1st, 3rd and 5th legs are given a sinus sign together, the joints of the 2nd, 4th and 6th legs having the same amplitude and frequency immediately after the contact of these legs to the ground sinus sign is applied. If the step angles on the 1st, 3rd and 5th legs are larger than the step angles on the 2nd, 4th and 6th legs, the robot moves in the *x* direction and vice versa. In the absence of trajectory control with PID and FLC controllers, the robot moves randomly with a tripod gait. The control mark of the controller controlling the movement in the *x* direction adjusts the pitch angles of the 1st, 3rd and 5th legs, and the control mark of the controller in the *y* direction adjusts the step angles of the 2nd, 4th and 6th legs. The control signal obtained from the PID and FLC controllers changes the amplitude of the sine signals providing tripod gait motion, thereby increasing or decreasing the pitch angle of the robot. When the trajectory is defined and this trajectory is given as reference to the PID and FLC controllers, the robot walks on this trajectory with wave velocity.

During the experimental simulation studies at Simulink, the weights of all parts of the robot were taken to be the same as the limits of the part weights actually planned. Experimental simulation studies were carried out in both control types and experimental values were obtained for motion on flat ground, motion on sloping ground, different weights and different friction coefficients. Based on the values obtained, FLC and PID controllers were compared.

A. Design of the hexapod robot

At this stage, parts are designed for robot formation in the dimensions, thicknesses and patterns determined in the SolidWorks design program environment. The designed robot has six legs. Inspired by spiders in nature, this six-legged chassis structure is structured in such a way that the legs are spread out in three parts to the right and left as spiders have leg and tail structure. Since the designed robot will include the camera structure and the elongated neck structure, a chassis design has been designed so that the legs are placed at 60 degree intervals in order to maximize the balance by ignoring the tail and leg portions. The preliminary drawing of the design is shown in Fig. 4.

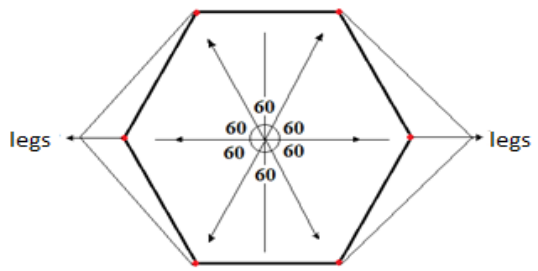


Figure 4. Chassis front drawing.

In the drawing of the frame, drawing between the legs of the robot was made to the right and left of each leg using a depth of 15 mm of a circle with a diameter of 73 mm. The plate thickness is 0.6 mm and the clearance between the two legs is designed as 94.5 mm. The process is shown in Fig. 5.

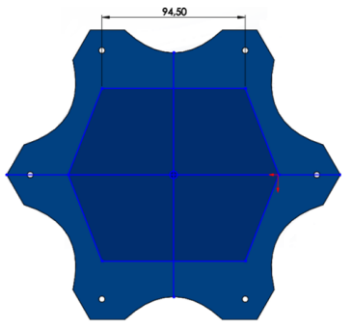


Figure 5. View of the drawn chassis.

A telescopic rod is placed at the center of gravity of the robot chassis. For the interior of the telescopic rod, 5 rollers are designed to engage one another and these rollers are associated with each other using a mate. The task of telescopic rod is to carry the camera. The camera is placed on the other end of this part. For the design of the camera, a solid model with an edge length of 22x22 mm was first created. The formed solid model was given a depth of 22 mm. The designed telescopic rod and camera is shown in Fig. 6 and Fig. 7. respectively.

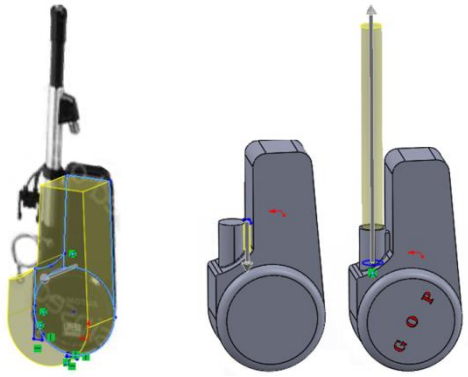


Figure 6. View of the drawn telescopic rod.

Figure 7. View of the drawn camera.

For servo motor rough drawing to provide leg movements, 40 mm height was given and 2 mm "radius" was applied to 4 corners. There are two servo motors on one leg of the robot. The first servo motor is attached directly to the chassis. The second servo motor is attached to the first servo motor by a leg member. The second servo motor has a leg member at the other end. The servo motor and leg design process is shown in Fig. 8 and Fig. 9 respectively.

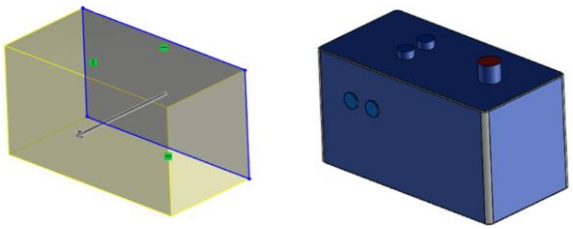


Figure 8. View of the drawn servo motor.

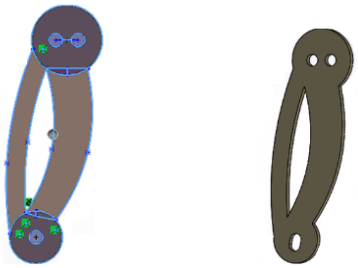


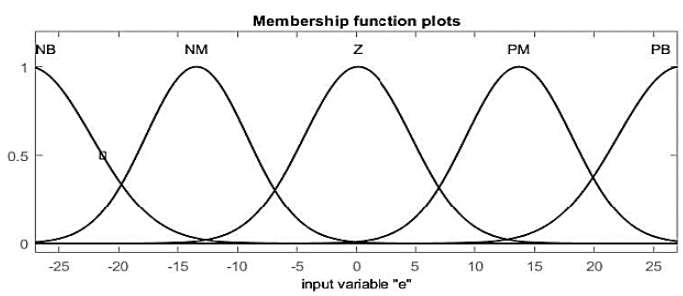
Figure 9. View of the drawn leg.

All designed parts of the designed robot are integrated into the assembly tab of the SolidWorks three-dimensional design program. This relationship differs according to the geometry of the contact points of the parts. These mates, which will directly affect the simulation control to be performed on the base of Simulink. Fig.10 shows the assembled robot.

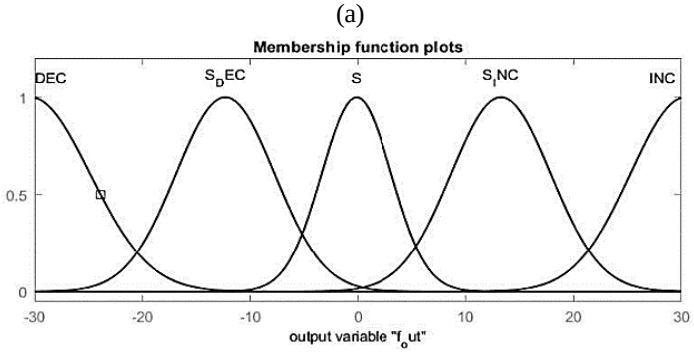
Figure 10. View of the assembled robot.

B. PID and FLC Design

In the FLC design, only the error (*e*) value was used as the input variable. The membership functions used for *e* and FLC output (*f_out*) are shown in Fig. 11. As seen, 5 gauss membership functions are used for *e* and *f_out*.



(a)



(b)

Figure 11. Membership functions used in FLC design.

The rules used for FLC are as follows. NB, NM, Z, PM and PB in these rules and in Fig. 11.a are abbreviations of Negative Large, Negative Medium, Zero, Positive Medium and Positive Large linguistic variables, respectively. DEC, S_DEC, S, S_INC, and INC to the right of the rules and in Fig. 11.b are abbreviations for the linguistic variables Reduce, Slightly Reduce, Wait, Slightly Increase, and Increase, respectively.

1. *If* (*e* is NB) *then* (*f_out* is DEC)
2. *If* (*e* is NM) *then* (*f_out* is S_DEC)
3. *If* (*e* is Z) *then* (*f_out* is S)
4. *If* (*e* is PM) *then* (*f_out* is S_INC)
5. *If* (*e* is PB) *then* (*f_out* is INC)

The same FLC was used for trajectory follow-up controls in the *x* and *y* directions. The outputs of both the *x*-direction and *y*-direction FLCs are multiplied by 50 gain value.

In the PID control experiments applied in the study, as in FLC experiments, the robot was controlled by two PID controllers on two axes including *x* and *y*. Integer order PID is used as PID controller. Adjustment of PID coefficients was done by trial and error method with repeated experiments. The PID coefficients used in the experiments are given in the table below. The outputs of the PID controllers are not multiplied by any gain coefficient (Tab.1).

Table 1. PID controller coefficients on the x and y axes.

Coefficients	PID*x*	PID*y*
K_p	10	14
K_i	0.5	0.4
K_d	5	5

IV. RESULTS

Trajectory must be created in order to send the robot to the desired and predetermined points within the working area. The trajectory graph for the robot is given in Fig. 12. According to the graph in Fig. 12, the robot must complete its movement in 150 seconds. It must pass through certain points on the *y*-axis within 150 seconds. The values in this graph constitute the *x* or *y* trajectory reference for the PID and FLC controllers that perform the orbit control of the robot. The motion trajectory of the robot on the *x*-axis is linear and the reference on the *x*-axis is zero for 0-150 seconds.

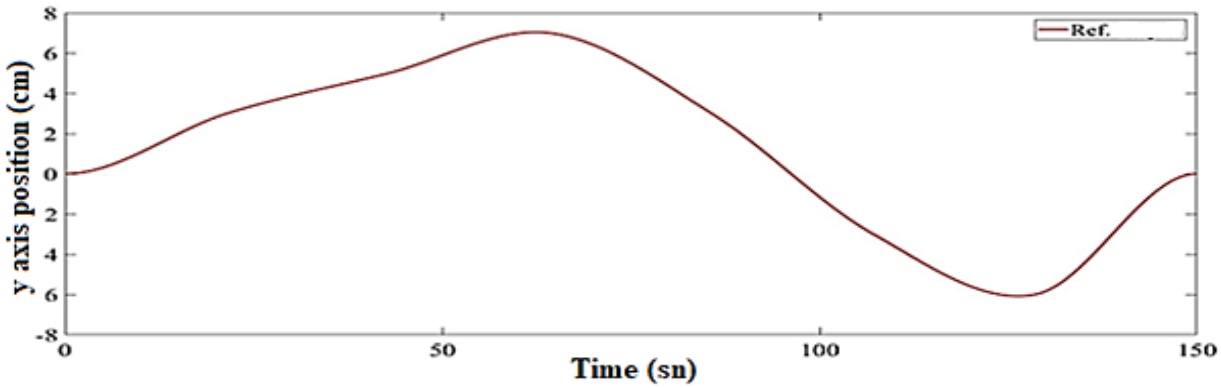


Figure 12. Robot reference trajectory graph.

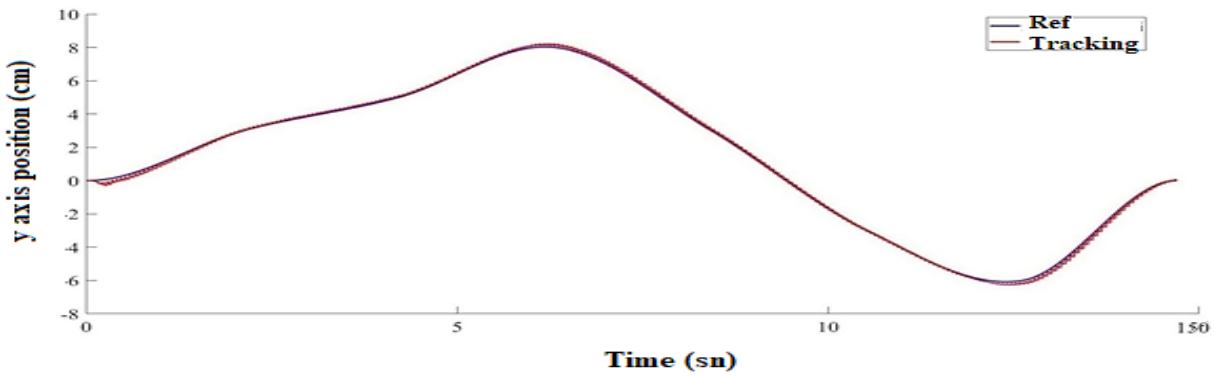


Figure 13. Trajectory graph for the PID controller.

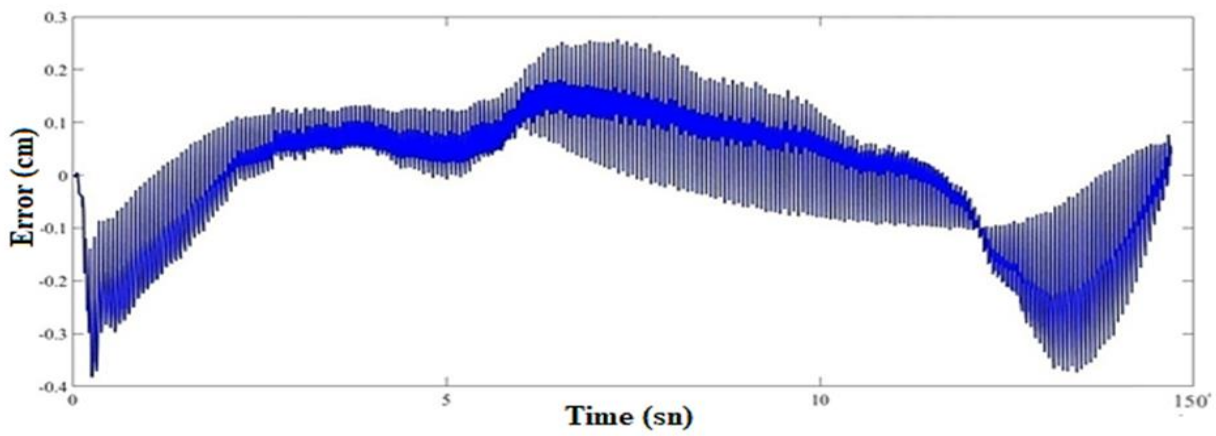


Figure 14. Trajectory tracking error graph on the *y*-axis for the PID controller.

Simulation experiments were first performed by applying PID controller to the robot and the graphs given below were obtained. The trajectory signal graphs generated on the *x* and *y* axes for the PID control method are given in Fig. 13-14.

When the Fig. 13 is examined, it is seen that the trajectory tracking error is higher in the rotation points on the trajectory (upper and lower peaks), but the PID controller can reduce this error in a short time.

For the PID controller, the error values obtained by taking the difference of the *y*-axis tracking signal generated as a result of the trajectory signal generated by the robot following the *y*-axis and the *y*-axis tracking signal are shown in Fig. 14. When the entire travel time is taken into consideration, the graph shows an error value of 0.3 cm initially generated by the robot in the process of capturing the reference signal. This is due to the friction of the robot feet with the floor and the inertia of the robot itself. Apart from this error, the average trajectory tracking error is seen to be 0.1 on average. This amount of error is a negligible range for a robot moving in terrain and tracking can be said to be a successful follow-up.

In order to better see the trajectory error of the PID controller, a cross section is taken on the trajectory graph on the *x* axis and this is shown in Fig. 15. Fig. 15 shows that the robot averages the signal towards the end of the said time. The reason for the undulating structure is that the robot is controlled by tripod gaits.

The difference of the reference signal and the tracking signal during the stage of controlling the robot on the *x* axis with PID controller is shown in Fig. 16. In the graph where the movement is shown completely, the error is seen to be within a change range of 0.4 cm. When this graph is analyzed, it is seen that the robot completes the tracking in the *x* axis with very negligible errors.

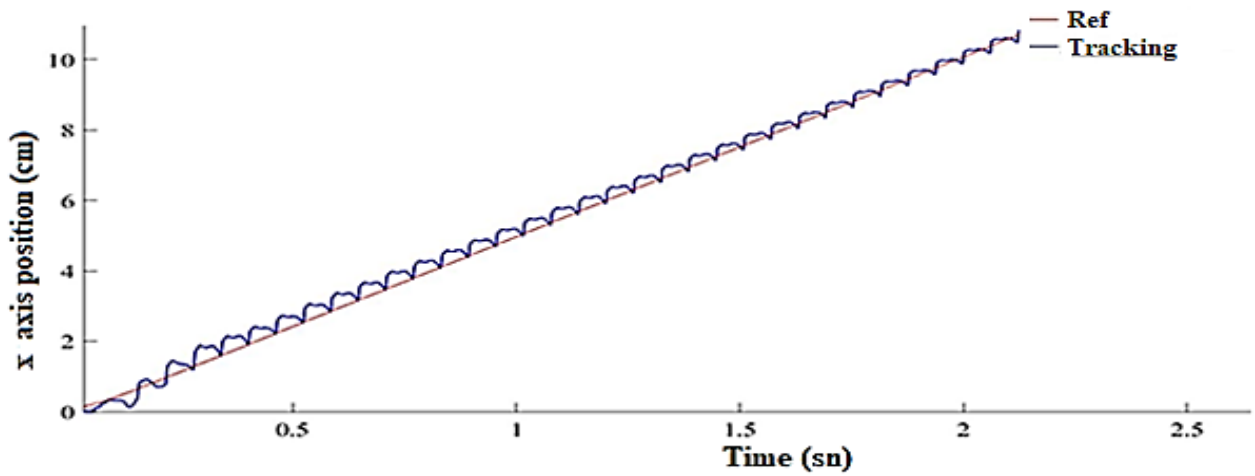


Figure 15. Trajectory graph on the *x*-axis for the PID controller.

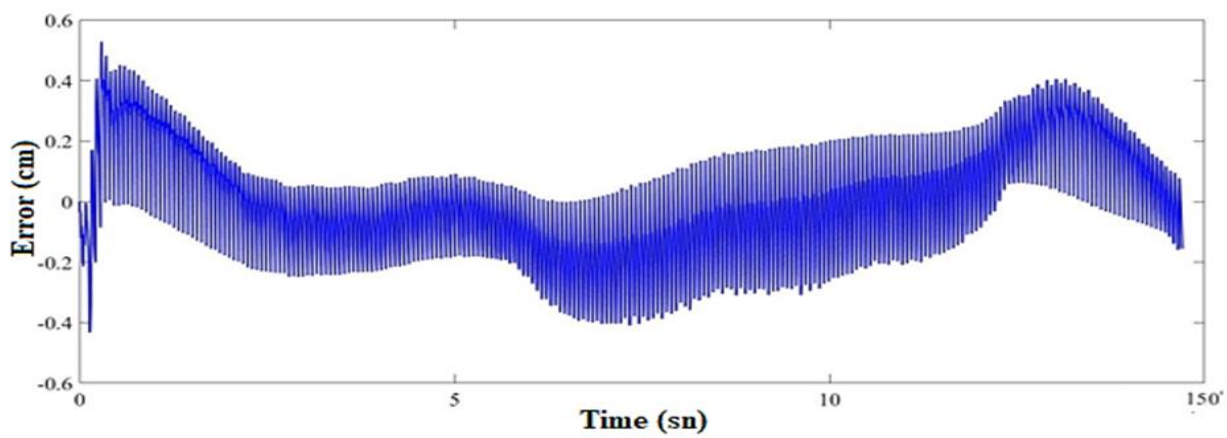


Figure 16. Trajectory tracking error graph on the *x*-axis for the PID controller.

The zoomed representation of the range of 25-29 sec during control of the robot's *x*-axis movement for the PID controller is shown in Fig. 17. From this graph, the reference tracking of the robot can be seen more clearly. The reason that the tracking signal is fluctuating is due to the robot's tripod gait motion.

Fig. 18 shows the trajectory tracking performance of FLC. According to Fig. 18, about 10-15 seconds after the start of movement, the robot was able to capture the reference signal and autonomously follow the reference signal. Acceptable trajectory tracking error was obtained along the trajectory.

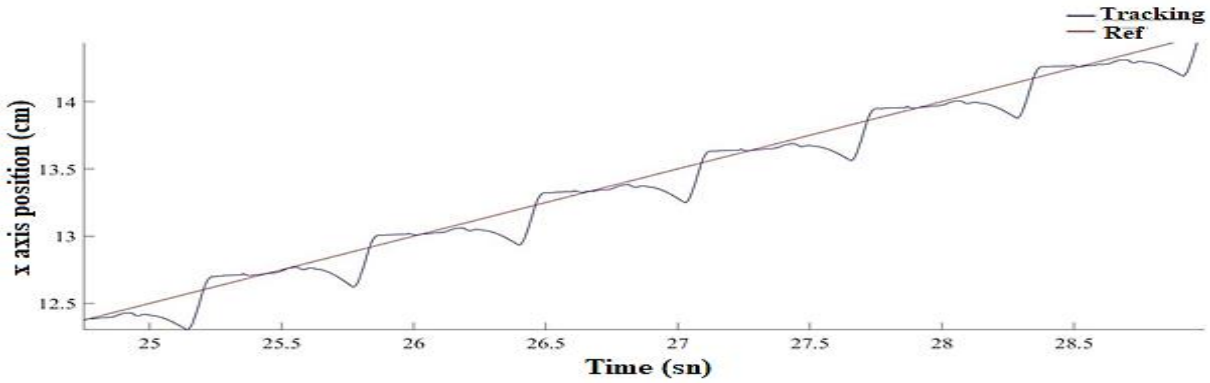


Figure 17. 25-29 sec interval of the reference-follow signal on the *x*-axis for the PID controller.

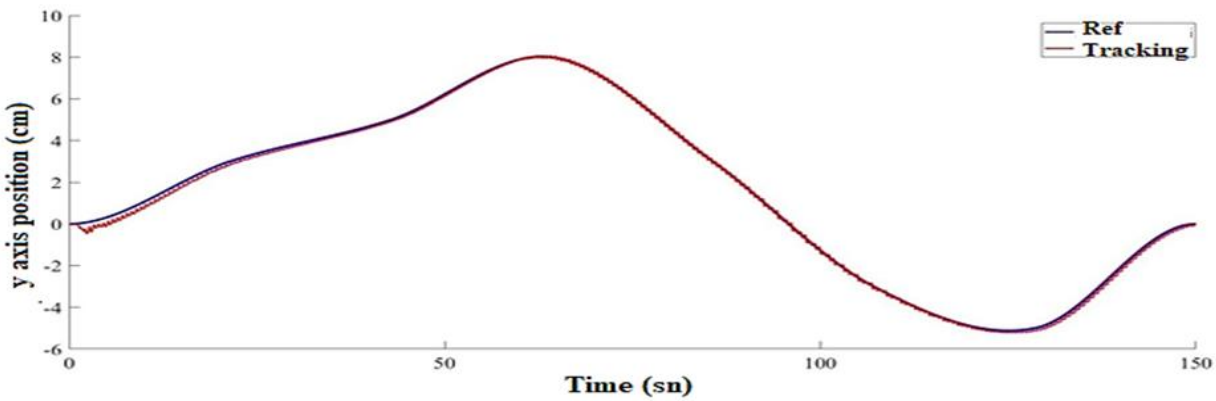


Figure 18. Trajectory graph on the *y*-axis of the FLC.

Fig. 19 shows the error signal on the y axis made with FLC. It appears that the signal initially oscillates to capture the reference signal and captures the reference value after 10-15 seconds. When the error change interval is examined considering the whole movement time, it is seen that the change range is between 0.1 cm and -0.1 cm. When the results obtained on the *y* axis of the PID controller are compared, it is seen that the error change interval of the FLC and the error change interval obtained with the PID controller are approximately the same. However, the error change interval at the turning points of the PID controller appears to be slightly larger than the FLC.

Fig. 20 shows the error signal made by the robot in the *x*-axis reference tracking with FLC. The error signal was obtained by taking the difference between the reference and follow signals. It is seen in the graph that the whole movement time is based on the error range between 0.2 cm and -0.2 cm. When the PID controller is compared with the *x*-axis error change graph, it is seen that there is more oscillation in PID controller performance than FLC results.

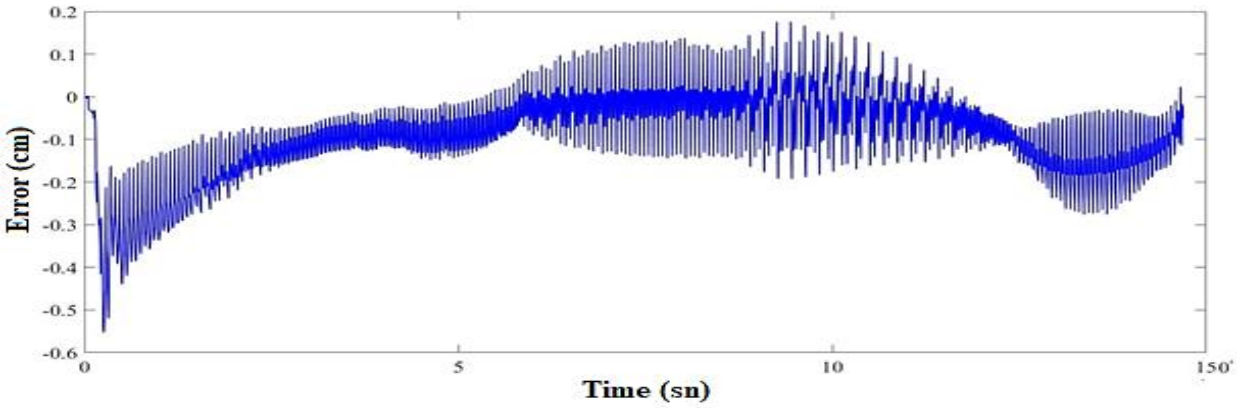


Figure 19. Reference trace error graph on the *y*-axis of the FLC.

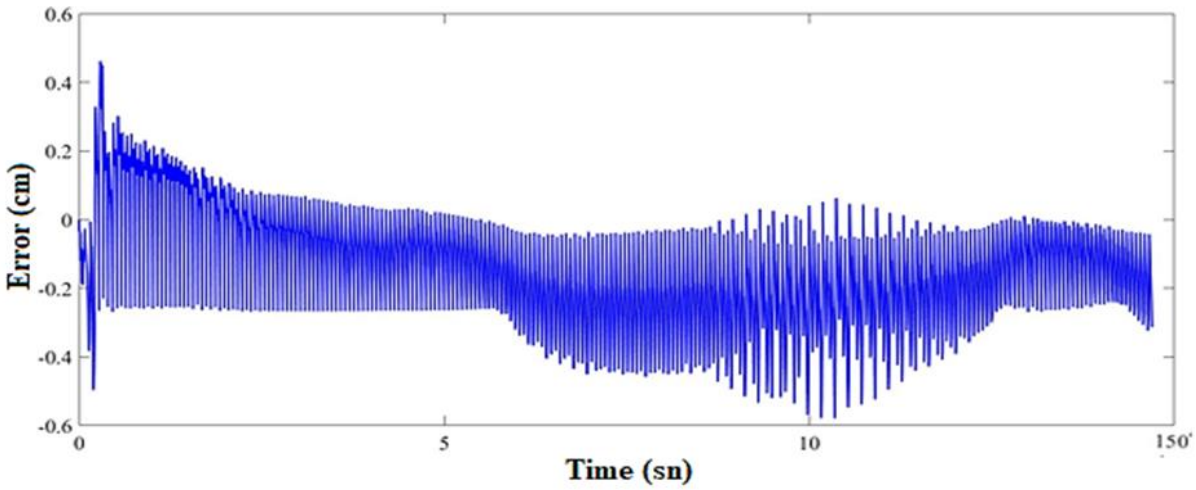


Figure 20. Error graph on the *x*-axis of the FLC.

V. Conclusion

In this study, CAD design of a six-legged spider robot was made in Solidworks program and these CAD designs were transferred to Simulink environment. Robot control simulation studies were performed in Simulink environment. The robot is intended to move autonomously on a particular trajectory. Robot motion trajectory is obtained by curve fitting methods. PID and FLC controllers were used for robot trajectory controls and robot trajectory motion performances of the two controllers were compared. In the study, during the controls, linear reference signal on *x* axis and an orbit on *y* axis were used in two types of controllers. Follow-up graphs were obtained during robot movement. In the analysis studies, the total weight of the robot with 4 kg was considered.

As a result of the analysis studies, it was observed that this robot, which was designed and analyzed, could follow a trajectory defined by a mass of 4 kg with an error value of approximately 1 mm. When FLC and PID performances are examined, it is concluded that the trajectory is followed with approximately the same error rates. During the trajectory tracking, the FLC controller performs better than the PID controller at the turning points of the reference trajectory curve of the robot. It would be useful to convert the design outputs to a real physical hardware and edit the motion control algorithms on a microcontroller system to create a physical prototype of the designed robot and to test the motion control methods under real environment conditions.

References

[1] G. Gürgüze, İ. Türkoğlu. "Kullanım Alanlarına Göre Robot Sistemlerinin Sınıflandırılması." Fırat Üniversitesi Müh. Bil. Dergisi 31(1), pp. 53-66, 2019.

[2] F. Tedeschi, G. Carbone. "Design Issues for Hexapod Walking Robots." Robotics 2014, 3, pp. 181-206; doi:10.3390/robotics3020181

[3] D. Chàvez-Clemente. "Gait Optimization for Multi-legged Walking Robots, with Application to a Lunar Hexapod." Ph.D. Thesis, Stanford University, California, CA, USA, 2011.

[4] G. Carbone, M. Ceccarelli. "Legged Robotic Systems. Cutting Edge Robotics" ARS Scientific Book, Vienna: pp. 553–576, 2005.

[5] S. E. Hamamci. "A New PID Tuning Method Based on Transient Response Control." Balkan Journal of Electrical & Computer Engineering, Vol.2, No.3, 2014 pp 132-138

[6] Caponetto, R., Fortuna, L., Porto, D. "Parameter Tuning of A Non-Integer Order PID Controller." 15th International Symposium on Mathematical Theory of Networks and Systems, Notre Dame, Indiana, 2002.

[7] A. Ates, B. B. Alagoz, G.T. Alisoy, C. Yeroglu and H. Z. Alisoy, "Fuzzy Velocity and Fuzzy Acceleration in Fractional Order Motion." Balkan Journal of Electrical & Computer Engineering, Vol 3, No 2, 2015, pp 98-102, DOI:10.17694/bajece.52354

[8] A. Sarabakha, C. Fu, E. Kayacan. "Intuitbeforetuning: Type-1 and type-2 fuzzy logic controllers." Applied Soft Computing, Vol 81, 2019. https://doi.org/10.1016/j.asoc.2019.105495

[9] O. Castillo, L. Amador-Angulo, J.R. Castro, M. Garcia-Valdez, A comparativestudy of type-1 fuzzy logic systems, interval type-2 fuzzy logic systemsand generalized type-2 fuzzy logic systems in control problems, Inform.Sci. 354 (2016) 257–274, http://dx.doi.org/10.1016/j.ins.2016.03.026.

[10] L. Cervantes, O. Castillo. "Type-2 fuzzy logic aggregation of multiple fuzzycontrollers for airplane flight control." Inform. Sci. 324 (2015) 247–256,http://dx.doi.org/10.1016/j.ins.2015.06.047.

[11] R.-E. Precup, H. Hellendoorn, "A survey on industrial applications of fuzzycontrol." Comput. Ind. 62 (3) (2011) 213–226, http://dx.doi.org/10.1016/j.compind.2010.10.001.

[12] A. Celikyilmaz, I.B. Turksen. "Modeling Uncertainty with Fuzzy Logic: With Recent Theory and Applications." first ed., Springer-Verlag Berlin Heidelberg, 2009, http://dx.doi.org/10.1007/978-3-540-89924-2.

[13] T. Kumbasar, H. Hagras. "Big Bang-Big Crunch optimization based interval type-2 fuzzy PID cascade controller design strategy." Inform. Sci. 282 (2014) 277–295, http://dx.doi.org/10.1016/j.ins.2014.06.005.

[14] C. Fu, A. Sarabakha, E. Kayacan, C. Wagner, R. John, J.M. Garibaldi. "Input uncertainty sensitivity enhanced nonsingleton fuzzy logic controllers forlong-term navigation of quadrotor UAVs." IEEE/ASME Trans. Mechatronic

BIOGRAPHIES

S. AYDIN received the B.S. degrees in mechatronic engineering from the University of Tokat Gaziosmanpasa University, in 2017. He currently studying M.S. degree in Tokat Gaziosmanpasa University, Graduate School of Natural and Applied Sciences from 2018. His research interests include robotic, CAD design of mechanical systems.

LEVENT GÖKREM was born in Tokat, TURKEY in 1970. He received the B.S. degree in Department of Electronics and Computer Education from Gazi University, Turkey, in 1993. He received the M.S. degree in Electronics and Computer Education from the Sakarya University Turkey, in 2003 and the Ph.D. degree in Industrial Technology Education from Gazi University, Ankara, in 2009. He has been an Assistant Professor with the Department of Mechatronics Engineering, Tokat Gaziosmanpasa University, Turkey. His research areas are Internet of Things, Image processing, robotic and wireless communication.

M. S. CAN received the B.S. degree in electrical-electronic engineering from the University of Nigde, in 2000 and M.S. degrees in electrical-electronic engineering from the University of Sutcuimam, in 2010 and the Ph.D. degree in electrical-electronic engineering from the University of Inonu, in 2017. From 2007 to 2018, he was a Lecturer in Zile Vocational School. Since 2018, he has been an Assistant Professor with the Electrical-Electronic Engineering Department, Tokat Gaziosmanpasa University. He is the author of 9 articles, 9 inventions. His research interests include automatic control applications, robotic, fuzzy logic, artificial neural networks.