Indonesian Journal of Electrical and Electronics Engineering (INAJEEE), Vol 8, No 1, 2025, page 9-16

Body Balancing Control System for Quadruped Robot While Walking on Uphill and Downhill Road Using Fuzzy

Ikrof Bayu Heka Rahmat[1*], M.Syarieffuddien Zuhrie[2]
[1,2] Electrical Engineering Department, Univeristas Negeri Surabaya
A5 Building Ketintang Campus, Surabaya 60231, Indonesia
[1]ikrofbayu.20051@mhs.unesa.id
[2]zuhrie@unesa.ac.id

Abstract – The development of robotics in Indonesia is growing rapidly, especially in the field of robotic legs. In the Indonesian SAR robot competition, there is an obstacle that is a sloping road that consists of a uphill road and downhill road. The problem with quadruped robots is the use of walking gaits that can only be used on flat fields, so they cannot be used directly on sloping roads. The researchers designed a Quadruped robot body balance control system using the fuzzy logic method, STM32F4 Discovery and Arduino nano as its controllers, the BNO055 IMU module as its sensor and the Dynamixel MX-28 servo for its legs. The result of this study was that the robot's time to reach a balanced body was a maximum of 1.18 seconds on a uphill road, with an equal error presentation of 2.5% roll angle and 0.95 seconds on the downhill road conditions, with a equal errors presentation of 2.2%. The test result of the high fit of the designed robot leg yielded an average error of 0.02 cm in front leg height on the uphill road, 0.02 in rear foot height when on the downhill road.

Keywords: *Four-legged robot, Uphill and Downhill Road, body balance, fuzzy logic.*

I. INTRODUCTION

The development of robotics in Indonesia is growing rapidly, especially in the field of legged robots. In response, the Ministry of Research, Technology and Higher Education of the Republic of Indonesia annually organizes the Indonesia Robot Competition (KRI) in which one of its categories is the competition of Indonesian fire fighter's robots. (KRPAI).[1]

At the Indonesian Firefighter Robot Competition (KRPAI) for this time the task of robots is only to extinguish candles, but by 2022, until 2024, the task changed to rescuing potential victims and changed its name to the SAR Indonesia Robot Contest. (KRSRI). In accordance with the name of the robot competition, which is to prioritize the task of the SAR team using a four-legged robot or Quadruped, then each track is given a more challenging obstacle, one of which is a sloping road consisting of a uphill and downhill road with a slope of approximately 16°. The robot is expected to be able to cross the uphill and downhill road without falling.[2]

Quadruped is a four-legged robot that consists of several servos and is able to regulate the movements of each leg. The use of a quadruped robot makes it easy for the robot to move freely and reach maximum range speed. However, there is still a problem with the use of the four-leged robot which has been developed that lies in the usage of walking gaits that are designed only for use on flat fields so that they cannot be used directly on sloping roads, because when walking on a slope obstacle, the gravitational style will make the robot easy to get out of control.[3] In addition, the robot's walking posture will cause the robot to slide, flip, or stumble, making it difficult to pass through the uphill or downhill road.[3] Therefore, there is a need for a dynamic balance control system of the body of the robot that can maintain a flat position as it passes through the trajectory and derivatives, so that the quadruped robot can stabilize its body in conditions where the COG (Center of Gravity) is constantly changing.

The research was carried out to design a system to control the balance of the body of a Quadruped robot as it passes through the uphill and downhill road using fuzzy logic methods. The fuzzy logic in this Quadruped robot is used to determine the input value of the inclination angle of the Roll variable, and then adjust the output to the servo rotation of the Dynamixel MX-28 to measure the height of the front and rear legs of the robot. By using the fuzzy logic method, the robot is expected to be able to balance its body remaining at the set point 0° even when walking on the roadside or derivative and have a better response than using other methods.

II. LITERARURE

Control System

A control system is an attempt to control, control and regulate an input with the aim of obtaining the desired output. There are two kinds of control systems: the open loop control system and the feedback control system. (Close loop system).[4]

An open-loop control system is a system whose output has no influence on the control action. outputs are not measured and are not reversed to be compared to inputs. When there is an interference in an open-loop system then the system will not perform the desired task.[5]

Figure 1. Open-loop control system
(Source: Dernoncourt, Franck, 2023)

Control System Closed-loop is also called a feedback control system. This sitem compares the actual output to the expected value and performs action based on the difference (error). Closed-loop control always uses feedback control actions to reduce system errors. [5] On Figure 2 is a close loop system diagram.

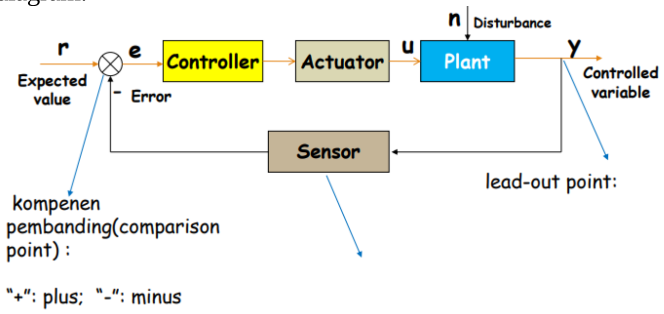

Figure 2. Closed-loop control
(Source: Dernoncourt, Franck, 2023)

Description:
1. Plant: controlled physical object
2. Controlled variable: variable controlled by the control system.
3. Expected value: the desired value or refraction value.
4. Controller: The component used for the control system.
5. Actuator: a mechanical device that converts any energy into motion.
6. Sensor: a device that measures physical magnitude and turns it into something that can be read by the controller.
7. Disturbance: an unexpected factor that disrupts the variable relationship within the system.

Quadruped Robot

A Quadruped robot is a type of legged robot that has four legs and is able to walk by controlling the movements of each leg. On this Quadruped robot there are three joints (3 DoF), where the most outer leg position of the robot is in contact with the floor, which allows the end-effector to move.[6]

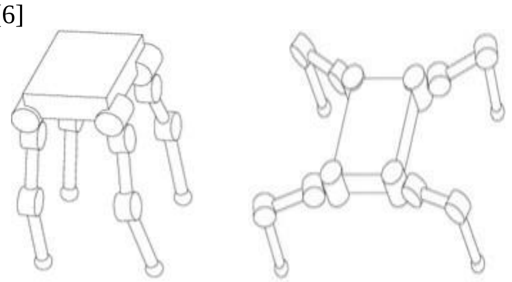

Figure 3. Mammal Type (Left), Sprawling Type (Right)
(Source: Kitano,dkk. Robomech J,2016)

A quadruped robot has two kinds of legs, the mammal type and the sprawling type. The mammal type is the leg of the thigh and thigh in a straight line vertically down, whereas in the sprawling type the leg is in a horizontal position and the legs are in a vertical position. [7]

STM32F4 Discovery

The STM32F4 Discovery is a microcontroller device made by STMicroelectronics that uses an ARM Cortex IC type STM 32F407VGT6. The board has speeds up to 168 MHz and is 32-bit ARM architecture microcontrollers. It has about 100 output input pins that have respective characteristics USART, UART, TIMER, ADC and I2C.

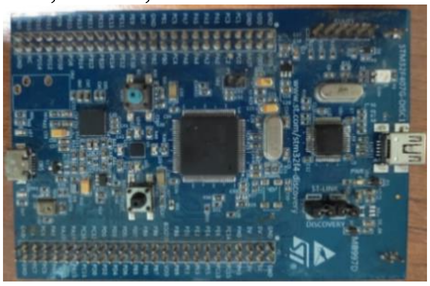

Figure 4. STM32F4 Discovery

The STM32F4 Discovery requires a working voltage of 3-5V DC, the programming language used on this microcontroller is the C language and the media used in this study is CooCox IDE software.[8] Table 1 below is the STM32F4 Discovery specification.

Table 1. STM32F4 Discovery Specification

Size	97 mm x 67 mm
Speed	168 MHz
I/O Pin	82
SPI	3
Working Voltage	3,3V – 5V
USART+UART Pin	4+2
ADC Pin	16

Arduino Nano

The Arduino Nano V3 is a small size board designed and manufactured by Gravitech. The board is designed using the Atmega328P type IC.[9] The Arduino Nano has 30 pins, eight of which (19-26 pins) are Analog pins. The other 14 pins are digital. [10] Arduino Nano in this study is used to process fuzzy data, Arduino nano specifications as follows:

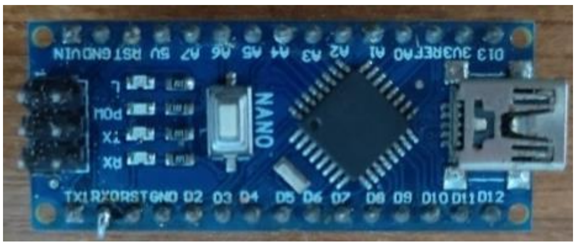

Figure 5. Arduino Nano

Table 2. Arduino Nano Specification

Microcontroller	Atmel ATmega328
Operating voltage	5 V
Input Voltage	7 - 12 V
I/O Analog and Digital pin	14 pin Digital and 8 pin Analog
Clock speed	16 MHz
Size	1,85cm x 4,3cm
DC Current I/O pin	40 mA
Flash Memory	32KB (ATmega328) 2KB

Dyamixel MX-28

Dynamixel MX28 is a type of servo made by robots in South Korea. Each servo has an ID that can be customized and modified as desired. The inner part of the servo uses full metal gear material and requires a voltage supply of 12V DC.[11] This Dynamixel servo engine is also equipped with UART, TTL, Multi Drop interfaces and has data communication speeds of 8000bps - 2Mbps.

Figure 6. Dynamixel MX-28
(Source: Haekal, 2016)

IMU BNO055

The IMU (Inertial Measurement Unit) BNO055 is a module that combines three sensors, the accelerometer, the gyroscope and the magnetometer. The 9 DOF combines 3-axis accelerometers, 3 axis gyroscopes, 3 geomagnetic sensors using the 32-bit ARM Cortex-M0 microcontroller in a single module and using a working voltage of 3.3V to 5V. The sensor works byining the 9-degree of freedom (DOF) which estimates the motion of position (X Y Z) as well as its orientation (roll, pitch, yaw) and the strength of the magnetic field along the axis. (X Y Z).[5]

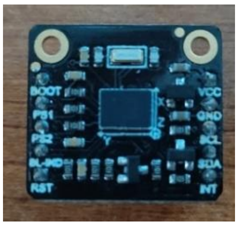

Figure 7. IMU BNO055 Sensor

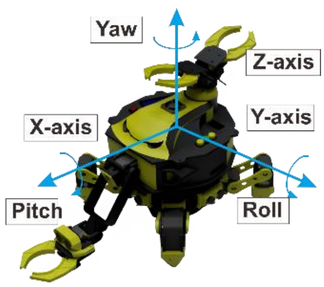

Figure 8. Axis Orientation on the Robot

These sensors can estimate the relative position, speed and acceleration of a movement. The output from this sensor is easily accessible by a microcontroller or a mini-pc via an I2C serial interface using a serial baud rate of 115.200 bps and an 0x50 (0x28 7 bit) address line with a clock of 400 kHz. It is fully compatible with Arduino and Raspy.[12] Here is an explanation of the three axes in Figure 8.

1. The Yaw
   Yaw axis is a condition when a robot turns left or right.
2. The Pitch
   Pitch Axis is the condition when the right or left side of a robot bends up or down.
3. The Roll
   Roll Axis are the conditions when the front or rear side of the robot bend to the top or the bottom.

Fuzzy Logic

Fuzzy logic is a branch of artificial intelligence, a knowledge that allows computers to mimic human intelligence so that it is expected that computers can do things that humans need to do. In other words, fuzzy Logic has a function to "simulate" the intelligence of humans to do something and implement it to a device, for example, robots, vehicles, household appliances, etc.

Fuzzy logic was introduced by Lotfi A. Zadeh, a professor of computer science at the University of California at Berkeley in 1965. When on ordinary logic only know right/wrong, yes/no, 1/0 but on Fuzzy logic allows the value between Yes/No, right/false, 1/0 can be defined. Comprehension is pretty fast and very fast can be formulated mathematically to get a more human way of thinking on computer programming.[13] Therefore, the concept of fuzzy logic corresponds to the human mindset that tends to judge an object vaguely. In the implementation of a fuzzy function on a device, there are three phases of the process: Fuzzification, Inference System, and Defuszification.

1. Fuzzyfication

   Fuzzyfication is a process of changing an entry from a definite form (crisp) to a fuzzy that is usually presented in the form of fuzzie sets with a function of their respective membership.

   Membership function (MF) or Membership Function is a graphical representation of the level of participation in each input, i.e. how each point in the input space is drawn to a value between 0 and 1. There are various forms of membership functions: triangular, trapezoidal, Gaussian, general bell curve, (π) type, and S type is the most common MF type.[14] However, in this study only two types of MF forms are used: triangle and trapezoid.
2. Fuzzy Inference System

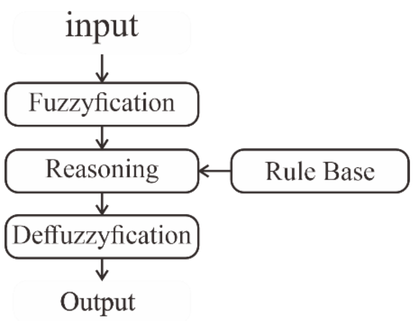

Figure 9. Fuzzy Logic System

   Fuzzy Inference System / FIS or fuzzy logic control system is a reference to describe the relationship between the input and output variables where the variables are processed and the resulting variables in a fuzzi shape. To explain the relation between inputs and outputs generally use "IF-THEN". One known type of FIS is Mamdani.[15]
3. Defuzzyfication
   Deffuzzification is the process of returning a fuzzy variable

from a composition of Fuzzy rules to certain data (crisp) that can be sent to the control device. The result of the defusification process is the output of the FIS. For FIS Mamdani, the process uses the centroid method or Centre of Gravity (COG).

III. METHODS

To design a control system on this research, it takes several stages to work on it. What steps are required can be seen in the following figure 10.

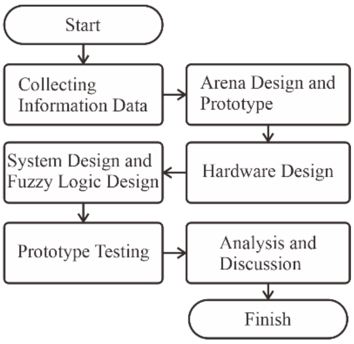

Figure 10. Research stages

A. Arena Design and Prototype

1. Arena Design

The design design of the arena was made for testing robots when walking on the uphill and downhill road. The arena is based on the rules of the Indonesian Robot Competition.

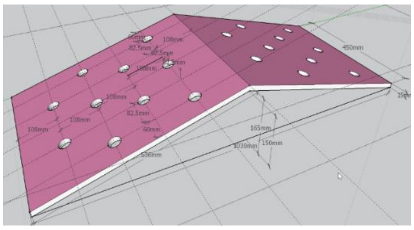

Figure 11. Uphill and downhill road arena

The dimensions used are 1030 mm long, 450 mm wide and 150 mm high.

2. Prototype

The design of the prototype used in this study had four legs with each leg three servos as it had one arm capita, two capita top, weighed about 3.2 kg, the body part used acrylic material with a thickness of 2 mm, the servo rider part and the inter-servos leg arm made of a plate of iron with 2 mm thick. The robot had dimensions of 27cm long, 27cm wide and 29cm high in on and standby position.

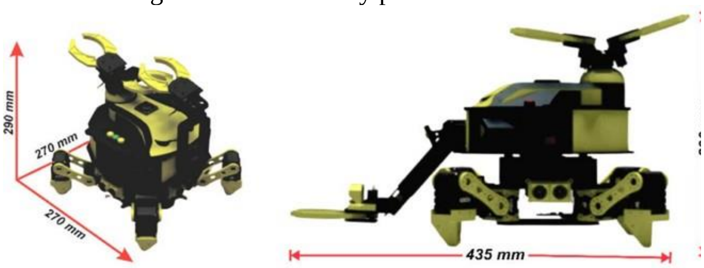

Figure 12. Prototype Design

B. Hardware Design

Figure 13 is hardware system block, the power source of the Quadruped robot uses a Lipo 12 VDC battery to activate the MX28 servo, in addition, the 12V battery voltage was previously reduced to 5V DC through the ubec for STM32F4 Discovery, Arduino Nano, IMU sensor module BNO055 and 16x2 LCD. The setpoint of the system is the desired degree value of 0°. STM32 and Arduino nano are used as control centers. The 16x2 LCD is used to display the measured angle values of the BNO055 IMU sensor module.

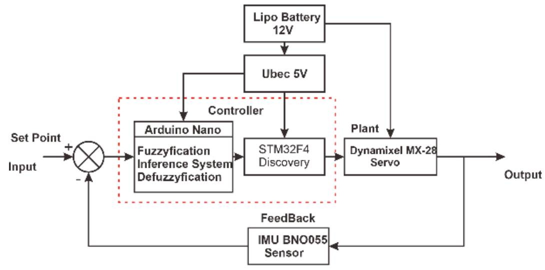

Figure 13. Hardware Design

C. System Design and Fuzzy Logic Design

1. System Design

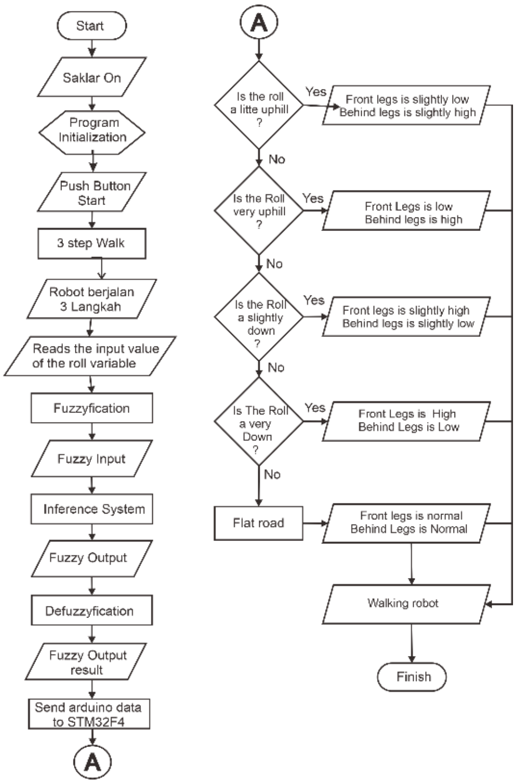

Figure 14. Flowchart System

The flowchart of how the system works begins with the terminator symbol as the beginning of the program of the system. Switch On to enable the robot, then go to the six-sided preparation symbol to initialize the program covering the declaration of the roll variable and the fuzzy library used. Next push button start is pressed, then the robot will run three steps first then the process of reading the Roll variable by the IMU sensor BNO055. After the angle of inclination is read by the sensor, the obtained roll data value will be the fuzzification input.

The process of fuzzification is the transformation of the numerical parameter number read by the sensor or the input value (crispt) into a number understood by the Fuzzy, namely 0 to 1. Furthermore, the fuzzy input precedes the process of the fuzzy inference system, which is the integration of variable values with the AND operator. As a result of the operation AND, the implication function used is MIN on each rule. After passing through the system inference process and the determination of base rule subsequently to the defuzzification process. Defuzzification is a process of transforming all the numbers collected into one and already processed previously into a crispt number. The defuzzification technique used is the centroid technique, this method searches for the Centre of Gravity (COG).

The whole process ranging from fuzzification to defuzzification is carried out automatically through the Arduino program using a fuzzy library, the result of the defuzzification process is sent to the STM32 via serial communication, where the result is to determine the set high position of the robot's legs while walking.

2. Fuzzy Logic Design.

In the study this time has one input variable that is the angle of the robot roll, as well as two output variables that are the height of the front leg of a robot and the heights of the behind leg of the robotic.

Figure 15 is an input and output design using a fuzzy logic designer in the MATLAB software version of 2017. Fuzzy input is obtained from the Roll variable with the degree unit resulting from the IMU BNO055 sensor reading.

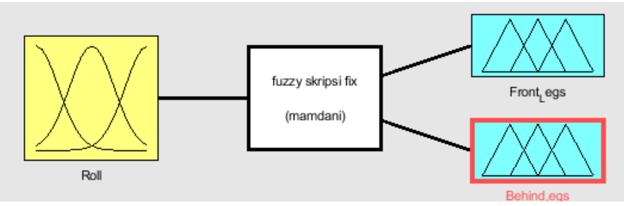

Figure 15. Input and Output fuzzy

Figure 16 is a fuzzy input of a roll variable that has a range from -20° to 20° and is divided into 5 categories of assembly, namely, very downhill, slightly downhill, flat, slightly uphill, very uphill. The very low category is in the range -20° to -9°. The slightly low category is within the range -10° to -2°. The flat category is in the range -2,5° to 2.5°, the slightly uphill category is within the range 2° to 10°, and the very uphill category is between 9° to 20°.

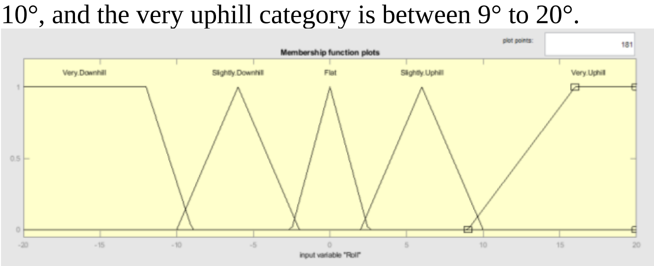

Figure 16. Membership Function Input roll

Figure 17 represents the membership function of the fuzzy output, which is the height variable of the front leg. Figure 18 is the membership function of the fuzzy output, namely the behind legs height variable. rear foot height is the same as the front, which has a range from 4 cm to 11,5 cm and is divided into 5 set categories. The low category is in the range of 4 cm to 4,7 cm. The slightly low category is in the range of 4,5 cm to 5,5 cm. The normal category is in the range of 5,2 cm to 7 cm. The slightly high category is in the range of 6,5 cm to 8,5 cm. The high category is in the range of 8 cm to 11,5 cm.

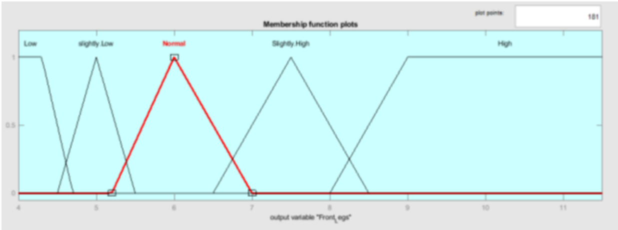

Figure 17. Membership Function output

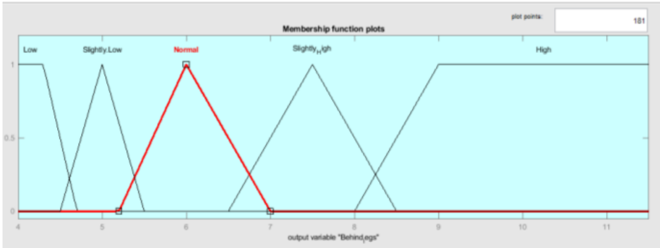

Figure 18. Membership Function output

The supporting theory in taking fuzzy logic requires data memorization. By looking for the error shown by the following equation (1) and (2)

Error = Roll_*Output* - Roll_Rf (1)

Where:

Roll_Rf = Reference Roll angle

Roll_Output = Output Roll angle

Example:

To find the error

Error = Roll_Output - Roll_Rf (2)

Error = 3 – 0

= 3

$$\text{Error\%} = \frac{|Roll\ Output - Roll\ Rf|}{Roll\ Rf} x\ 100\%$$

Description:

| RO – RRf |= Absolute result RO - Roll reference

Error% = 3 %

Explanation of IF-THEN rules that have been created in MATLAB

a. If (Roll is very downhill) then (front legs is High) (behind legs is Low) (1).
b. If (Roll is Slightly downhill) then (front legs is Slightly High) (behind leg is Slightly Low) (1).
c. If (roll is flat) then (front legs is Normal) (behind legs is Normal) (1).
d. If (roll is Slightly Uphill) then (front legs is Slightly Low) (behind legs is Slightly High) (1).
e. If (Roll is Very Uphill) then (behind legs is Low) (behind legs is High) (1).

IV. RESULT AND DISCUSSION

A. Robot Testing Results without Fuzzy Logic

In this test, the robot will be tested passing uphill and downhill roads without using fuzzy logic. The experiment was carried out once in each arena, the data observed was the roll angle value displayed on the LCD and the Arduino software serial monitor when the robot passed uphill and downhill roads.

Figure 19 is a graph made on Microsoft excel data streamer results from reading the Arduino software serial monitor every 10 ms, showing that the roll value increases slightly until the position of the robot body is parallel to the ramp and the roll variable is at a maximum value of 21°.

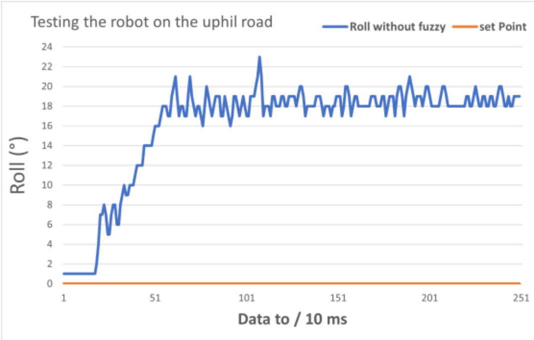

Figure 19. The graphs of the robot's road test results without fuzzy

The picture of the robot test results can be seen in Figure 20. When the robot's posture follows an uphill road, it can cause the robot to tip over.

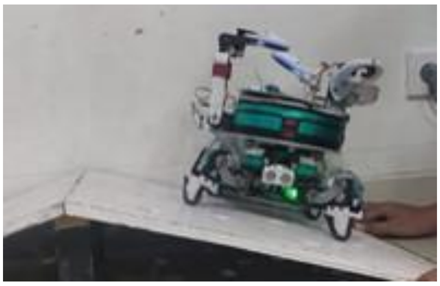

Figure 20. Robot on the uphill road

Figure 21 is one of the graphs of the robot's road test results without fuzzy logic when walking through a downhill road. The data reading is the same as the uphill road, which is every 10 ms, the graph shows that the roll value decreases little by little until the position of the robot body is parallel to the downhill road and the roll variable value is at a maximum value of -18 °.

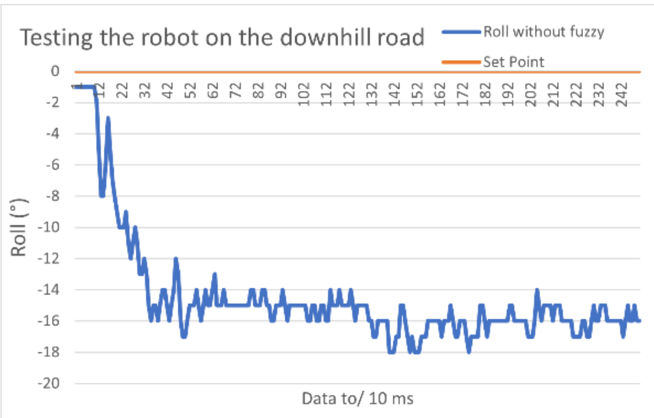

Figure 21. The graphs of the robot's road test results without fuzzy

The robot's body follows a downhill road, so the robot easily tumbles forward. The picture of the robot test results can be seen in Figure 22.

Figure 22. Robot on downhill road

B. Robot Testing Results with Fuzzy Logic

Experiments were carried out 7 times each, the data observed were the maximum roll angle value displayed and the robot's time response in balancing its body. The roll angle data reading of 10 ms is observed through the LCD and Microsoft excel data streamer when the robot passes uphill and downhill roads. the test results will show a comparison of the roll angle graph between using fuzzy logic and without fuzzy logic.

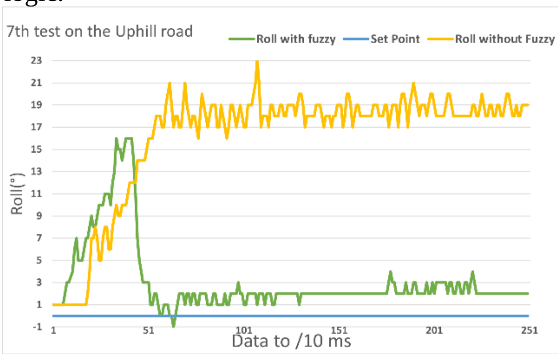

Figure 23. The result graph on the uphill road test

Figure 23 is a graph of one of the results of the 7th test uphill road test, roll comparison when using fuzzy (green color graph), not using fuzzy (yellow color graph), set point (blue color graph). The robot response or change in roll angle approaching the average set point occurs in the 70th data to the 75th data since the robot start push button is pressed, this indicates that the robot position is in a balanced state because the height of the foot adjusts to the slope of the uphill road.

Table 3. Results of testing the robot on the uphill road

No	Set Point (°)	Read Roll Angle (°)	*Error* (°)	Time Respone (second)
1	0	3	3	1,1
2	0	2	2	1,2
3	0	2	2	1,1
4	0	2	2	1,15
5	0	3	3	1,26
6	0	2	2	1,3
7	0	4	4	1,2
Average			2,5	1,18

Table 3 is a test result that shows that the Quadruped robot can balance its body with an average maximum roll angle reading error of 2,5° with an average response time of

1,18 seconds since the robot tilt angle changes from 16° to 0°. The picture of the robot test results can be seen in Figure 24.

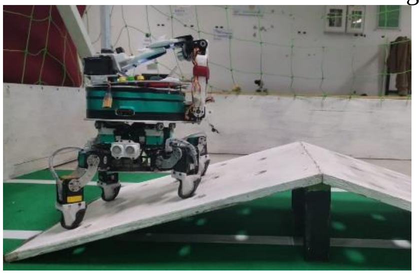

Figure 24. Results of testing the robot on the uphill road

Next is the result graph on the downhill road test and roll comparison when using fuzzy (green color graph), not using fuzzy (yellow color graph), set point (blue color graph).

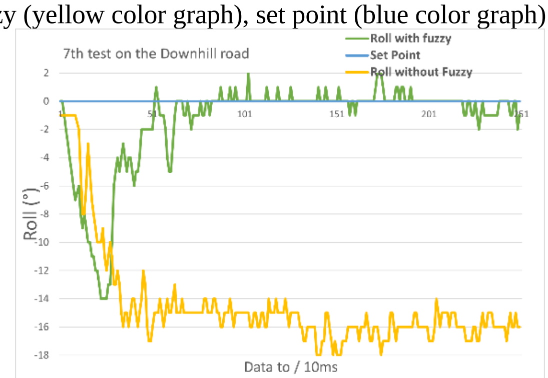

Figure 25. The result graph on the downhill road test

The robot response or change in roll angle approaching the average set point occurs in the 70th data to the 75th data since the robot start push button is pressed, this indicates that the robot position is already in a balanced state because the height of the foot adjusts to the slope of the descent road. Next is a table of test results for 7 times.

Table 4. Result of testing the robot on the downhill road

No	Set Point (°)	Read Roll Angle (°)	*Error* (°)	Time Response (Second)
1	0	2	2	1
2	0	3	3	1,1
3	0	3	3	1
4	0	2	2	1,1
5	0	2	2	1,2
6	0	2	2	1,2
7	0	2	2	1,2
Average			2,2	0,95

The test results in Table 4 show that the Quadruped robot can balance its body with an average maximum roll angle reading error of 2,2° with an average response time of 0,95 seconds since the robot tilt angle changes from -15° to 0°. The picture of the robot test results can be seen in Figure 26.

Figure 26. Robot on downhill road

C. Fuzzy Logic and Time Testing Results

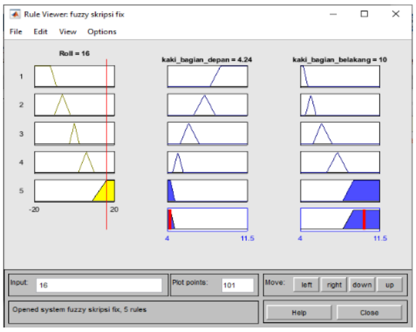

Figure 27. Rule viewer in MATLAB

Figure 27 is a fuzzy result based on the rule base that has been created in MATLAB, the input value of the roll variable is 16° at an uphill angle, and produces an output in the form of a front foot height value of 4,24 cm and a rear foot height value of 10 cm.

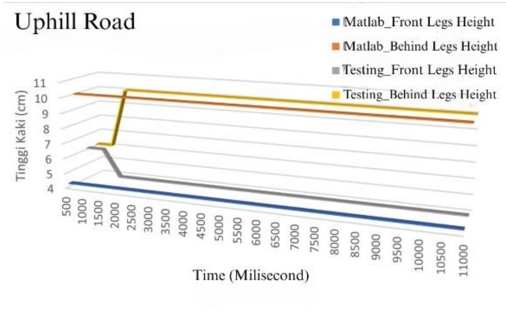

Figure 28. Graph of the results of testing the suitability

Figure 28 is a graph of the test results, based on MATLAB for the front leg height (blue color graph), MATLAB for the rear leg (orange color graph), based on testing for the front leg (gray color graph), testing for the rear leg (yellow color graph).

Table 5. Results of testing the robot on the uphill road

No	Front legs height (cm)	Behind legs Height (cm)	Travel time (second)
1	4,26	10	10,83
2	4,26	10	9,90
3	4,26	10	11,20
4	4,26	10	10,78
5	4,26	10	10,80
Average			10,7

The test results in Table 5 show that the height value of the front leg is not in accordance with the fuzzy results in MATLAB, which is 4,26 cm and there is an error of 0,02 cm. while the value for the back leg is in accordance with the fuzzy results in MATLAB, which is 10 cm. while the average travel time for the robot to successfully pass the uphill road is 10,7 seconds.

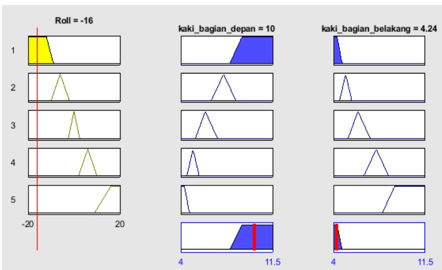

Figure 29. Rule viewer in MATLAB

Figure 29 is a fuzzy result based on the rule base that has been created in MATLAB, the input value of the roll variable is -16° at the downward slope angle, and produces an output in the form of a front foot height value of 4,24 cm and a rear foot height value of 10 cm.

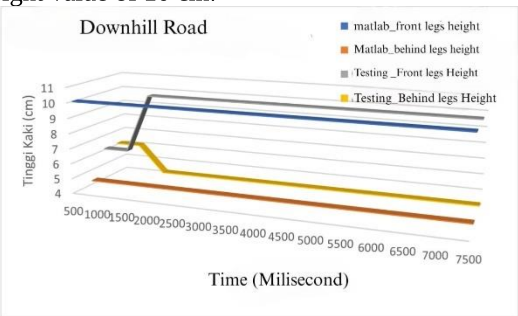

Figure 30. Graph of the results of testing the suitability

Figure 30 is a graph of the results of testing the suitability of the robot leg height, based on MATLAB for the front leg height (blue color graph), MATLAB for the back leg (orange color graph), based on testing for the front leg (gray color graph), testing for the back leg (yellow color graph).

Table 6. Results of testing the robot on the downhill road

No	Front Leg Height (cm)	Behind legs Height (cm)	Travel time (second)
1	10	4,26	7,20
2	10	4,26	6,82
3	10	4,26	7,50
4	10	4,26	7,81
5	10	4,26	6,93
Average			7,25

Based on the test results in Table 6, it shows that the height value of the hind legs is not in accordance with the fuzzy results in MATLAB, which is 4,26 cm or there is an error of 0,02 cm. while the value for the back leg is in accordance with the fuzzy results in MATLAB, which is 10 cm. while the average travel time for the robot to successfully pass the downhill road is 7,25 seconds.

V. CONCLUSION

Based on the research and testing that has been carried out, the following conclusions are obtained:

a. The body balance control system on the Quadruped robot was successfully designed and made by applying fuzzy logic as its control. This is indicated by the suitability of the height of the robot's legs designed in MATLAB with the test results, which produces an average error of 0,02 cm in the height of the front leg when on an uphill road, and has an average travel time of 10,7 seconds. while 0,02 cm in the height of the rear leg when on a downhill road and has an average travel time of 7,25 seconds.

b. The fuzzy logic system implemented has worked quite well even though there is a slight error in the reading of the IMU BNO055 sensor when walking on uphill and descents of a maximum of 2,5° due to the oscillation of the robot when walking, but the height of the robot's legs can adjust based on the roll angle value on uphill and downhill road.

c. With this test, the response time of the robot to reach a balanced body is a maximum of 1,18 seconds on an uphill road condition of 16 ° with an average percentage of roll angle error of 2,5%, and 0,95 seconds on a downhill road condition of -15 ° with an average percentage of roll angle error of 2,2%. There are several factors that prevent the robot from passing uphill or downhill roads, namely slippery road surfaces, the shape of the robot footwear used, the weight of the robot.

ACKNOWLEDGMENT

Thank you to Dr. M. Syariffuddien zuhrie, S.Pd., M.Pd., as the supervisor who has provided valuable guidance and input throughout this research.

REFERENCES

[1] B. Kusumoputro *et al.*, “Panduan Kontes Robot Indonesia (KRI) year 2022,” p. 146, 2022.

[2] B. Kusumoputro *et al.*, “Buku Pedoman Kontes Robot Indonesia (Kri) year 2023,” pp. 1–150, 2023.

[3] Z. Nasution, A. F. Ilham Suparman, G. A. Prasetyo, A. H. Alasiry, E. H. Binugroho, and A. Darmawan, “Body Balancing Control for EILERO Quadruped Robot while Walking on Slope,” *IES 2019 - Int. Electron. Symp. Role Techno-Intelligence Creat. an Open Energy Syst. Towar. Energy Democr. Proc.*, pp. 364–369, 2019.

[4] M. O. Turakulovich, “International Multidisciplinary Journal For Research & Development And Installation Of An Automatic Temperature,” vol. 10, no. 12, pp. 2020–2024, 2023.

[5] I. Kuric, “Inclination Stability Controller For Spot Micro Robot Based on Artificial Neural Networks,” no. December, 2022.

[6] R. Cahyo Prayogo, A. Triwiyatno, and D. Sumardi, “Perancangan Robot Berkaki 4 (Quadruped) Dengan Stabilization Algorithm Pada Uneven Floor Menggunakan 6-Dof Imu Berbasis Invers Kinematic,” *Transient J. Ilm. Tek. Elektro*, vol. 7, no. 2, pp. 543–551, 2018.

[7] H. E. Saputro, N. A. S. Putro, S. Hartati, and I. Usuman, “Implementasi Sistem Kendali Keseimbangan Statis Pada Robot Quadruped Menggunakan Reinforcement Learning,” *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 13, no. 1, 2023.

[8] J. Billingsley and J. Billingsley, “Implementasi Inverse Kinematics Pada Pergerakan Robot Quadruped,” *Its*, pp. 1–110, 2017.

[9] M. ALFAYET, “Perancangan Sistem Pengontrolan Pengisian Air Siap Minum Berbasis Arduino (Studi Kasus Bintang Asih),” 2022.

[10] R. Ranjane, R. Chavan, and R. Mahajan, “Quadruped Robot with Speed and Direction control,” *Int. Res. J. Eng. Technol.*, vol. 08, no. 01, pp. 1965–1969, 2021.

[11] M. Haekal Maulana, A. Arifin, and R. Dikairono, “Design of Walking Movement on Bipedal Antropomorphic Robot Using Inverse Kinematics and Gait Data,” 2016.

[12] Bosch Sensortec GmbH., “BNO055 Data sheet,” no. November, p. 105, 2014.

[13] M. Hellmann, “Fuzzy Logic Introduction,” no. 1, 2001.

[14] M. D. S. Y. P. SOCIAL, “Structural Analysis of Covariance on Health-Related Indicators among the Elderly at Home, Focusing on Subjective Health Perception,” *Appl. Microbiol. Biotechnol.*, vol. 85, no. 1, pp. 2071–2079, 2014.

[15] A. Varshney and V. Goyal, “Re-evaluation on fuzzy logic controlled system by optimizing the membership functions,” *Mater. Today Proc.*, 2023.