Design of Barelang F.1 Robot Using Genetic AlgorithmMethod for Multiplication OnUneven Floor

Diono[1], Yoel Panangian Tambunan[2], Fadli Firdaus[3], Adlian Jefiza[4]

{diono@polibatam.ac.id[1], yoelpanangiantambunan123@gmail.com[2], fadlifirdaus@polibatam.ac.id[3], adlianjefiza@polibatam.ac.id[4]}

Politeknik Negeri Batam, Electrical Engineering Department, Indonesia[1,2,3,4]

Abstract. A quadruped robot is a legged robot that has four legs to move. Barelang F.1 isa robot that moves with inverse kinematics, with inverse kinematics this robot can move by giving a value to each joint on robot's legs [1]. Barelang F.1 is a robot that we built to compete in the Indonesian Robot Contest (KRI) event. In that competition, we find that there are still several challenges that our robot can't handle. Uneven obstacles and inclined planes will be a challenge for the Barelang F.1 robot now because we still don't have a method to overcome that challanges. The purpose of making this journal is to conduct research on these existing problems and find solutions for that. So we choose the genetic algorithm method to see if this method will be the best solution to our robot to overcome the problems that we encounter. After a several test we find that the result from genetic algorithm method it had a small error that can make the robot to stabilize itself but for that small error it took a long time to process, so this method can be applied to find the best angle for the robot to stabilize itself but it can used during the competition due to the time that it takes to proceed the calculation.

Keywords: Genetic Algorithm, Stabilization, Robotic Movement

1. Introduction

Industrial progress has entered the 4.0 era can be seen by several research institutes conducting technological research in various fields [2]. The technology that is developing rapidly at this time is technology in the field of robotics [3]. Many examples of robots have been developed, ranging from legged robots to flying robots and wheeled robots. A quadruped robot is a four-legged robot that can walk by adjusting the movements of each leg [4]. The use of legged robots is generally for exploring the outdoors on military missions, exploring forests, and it is used for research on animal behavior and testing artificial intelligence [5].

Barelang F.1 is one of the robot teams joined in BRAIL for the KRSRI (Kontes Robot SAR Indonesia or Fire Fighting Robot Competition) robot division. KRSRI is a legged robot competition contested at the Indonesian Robot Contest (KRI). KRSRI is a fire-fighting robot competition held in Indonesia. Trinity College Fire Fighting Home Robot Contest is an international-level fire-fighting robot competition [4]. For the 2022 KRSRI competition, there are several obstacles, one of which is an incline obstacle with a slope of 15° and the drawback of the quadruped robot is that when it faces an uneven or sloping floor surface (uneven floor),

ICAE 2023, November 07, Batam, Indonesia

DOI 10.4108/eai.7-11-2023.2342947

the robot will walk unstable [6].

This challenge is a tough one for our robots at this time. The Barelang F1 robot needs a method to overcome these obstacles. Things that might affect the final result include field conditions and robot mechanics.

To solve this, studied was conducted to complement the shortcomings of the Barelang F.1 robot so that the robot can compete in future competitions. This study used a quadruped robot with 12 servos divided into three servos for each leg. Each leg has 3 DOF (Degree of Freedom).Trial will be tested on a surface with a slope of 15°. The data output from the imu sensor will be processed by the genetic algorithm in order to get the best position for the robot to pass through the obstacle.

2. Method

2.1. Hardware Design

The design of the mechanical specifications of the Barelang F.1 robot adapts a quadruped robot with four legs and uses 12 DoF (Degree of Freedom) divided by one leg by 3 DoF for propulsion towards the destination coordinates. Legs designing for robots needed to maximize the movement of robots to make it more dynamic [7]. The Barelang F.1 robot has a leg size with a coxa link length of 21.70 mm, a femur link of 50.08 mm, and a tibia link of 76.95 mm. From the specifications above, it can be seen in Fig. 1.

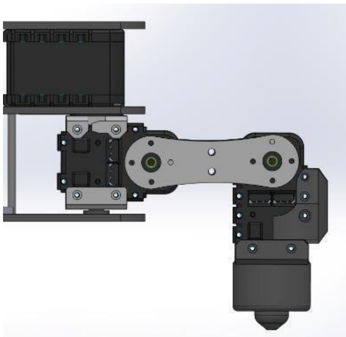

Fig. 1. Barelang F.1 robot leg design

The Barelang F.1 robot uses Raspberry Pi 4 as a microcontroller for sensors and also as the core of the microcontroller and OpenCR as the AX-18A servo controller. The power of this robot uses a 3-cell LiPo battery with a current of 5000mAh and a voltage of 11.1V. The hardware design can be seen in Fig. 2.

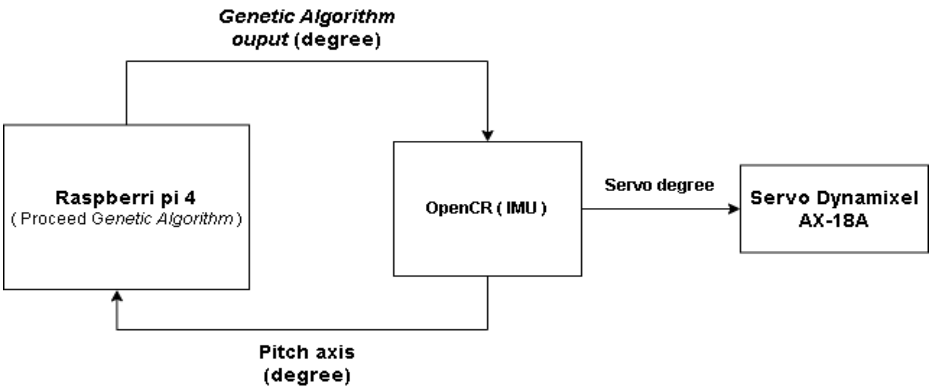

Fig. 2. Barelang robot Hardware block diagram

2.2. IMU MPU9250 Sensors

The Inertial Measurement Unit (IMU) is a tool that utilizes measurement systems such as gyroscopes and accelerometers [8] that used to determine the whereabouts and movement of an object [9], to estimate the relative position, speed and acceleration of motor movements. The accelerometer is used to measure the acceleration of an object and the gyroscope to measure the rotation of an object. The IMU maintains the position of an object in motion with the x, y, z axes and orientation (roll, pitch, yaw). The IMU sensor used in this study is the MPU9250 found on opencr.

2.3. Genetic Algoritm

Genetic algorithm is a method used to solve optimization from simple problems to complexs [10]. A genetic algorithm is an algorithm that adapts the natural evolutionary process to the concept that the most superior genes or individuals survive and the weak individuals will perish. The superiority of each genes will be tested through a process called the fitness function by maximizing the existing compatibility between individuals and later be called fitness value [11]. The genetic algorithm process has several steps, namely:

- Initial population

  In this section, a random individual solution is form and soon called the population. The number of the population depends on the level of difficulty of the problem itself. But usually, the number of population can reach hundreds or even thousands of solutions to the problem [12]. Then the populations tested with the fitness function to find the fitness value of the population.

- Selection

  In this section, the best candidate solution will be selected based on the fitness value that has been calculated for each gene or individual and will be ranked from the best to the worst gene. This part is to produce good parents. By getting good parents, the offspring will also be good with high fitness value [13].

- Reproduce
  - Crossover

    In this section, the best candidate that been selected will be used as the parent. Then the genetic exchange of the parents is carried out and it will produce new offspring that containthe genes from the parents earlier that can be called child.
  - Mutation

    In this section, the existing child will have its genes mutated to form a new gene that has the genes from the previous parents but also has differences from the previous gene because of the mutation process earlier.
- Termination

  The above process will be repeated until a termination condition is found [12]. The conditions are:
  - A solution meets the minimum criteria

    The solution with the highest match value has been reached

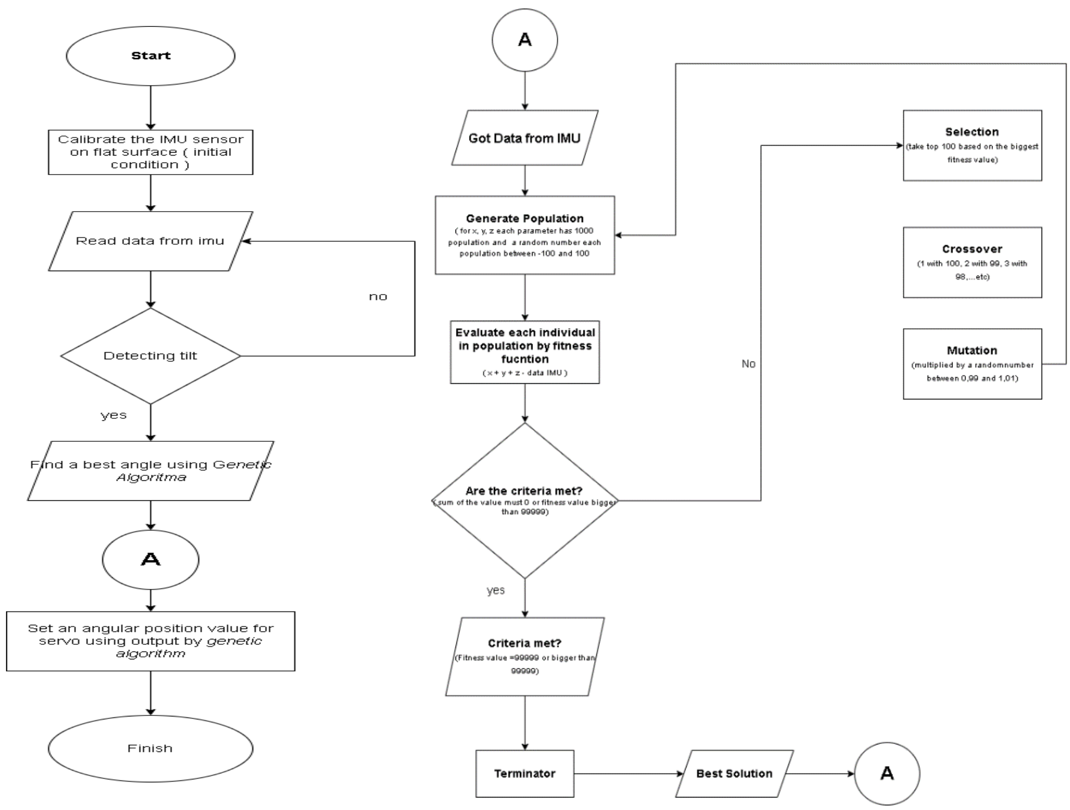

Fig. 3. System flowcharts

3. Results

3.1. MPU9250 Sensor Angle Testing

Testing the MPU9250 sensor is done by giving angle values, namely 0 degrees, -15 degrees, 15degrees, -17 degrees, and 17 degrees. The trial is tested on a quadruped robot pitch orientation angle.

3.2. Pitch Angle Testing on IMU

Table 1. Degree reading with MPU9250

Degree reading with MPU9250						
No	**-17°**	**-15 °**	**0 °**	**15 °**	**17 °**	**Error**
1	-17	-15	0	15	17	0%
2	-17	-15	0	15	17	0%
3	-17	-15	0	15	17	0%

3.1.1. Test result

Genetic algorithm tested on the Barelang F.1 robot was done on several variations of slope, namely slope 0°, -15°, 15°, -17°and 17°. In this paper, the error results in a genetic algorithm are obtained by subtracting the results from a genetic algorithm with an initial set. To find real results on robots using the Protractor application. The following table testresults.

3.1.2. Test result

Genetic algorithm tested on the Barelang F.1 robot was done on several variations of slope,namely slope 0°, -15°, 15°, -17°and 17°. In this paper, the error results in a genetic algorithm are obtained by subtracting the results from a genetic algorithm with an initial set. To find real results on robots using the Protractor application. The following table testresults.

- Data collection using the Protractor application.

Fig. 4. Angle measurement with the Protractor application

- Graph of genetic algorithm results at 0°

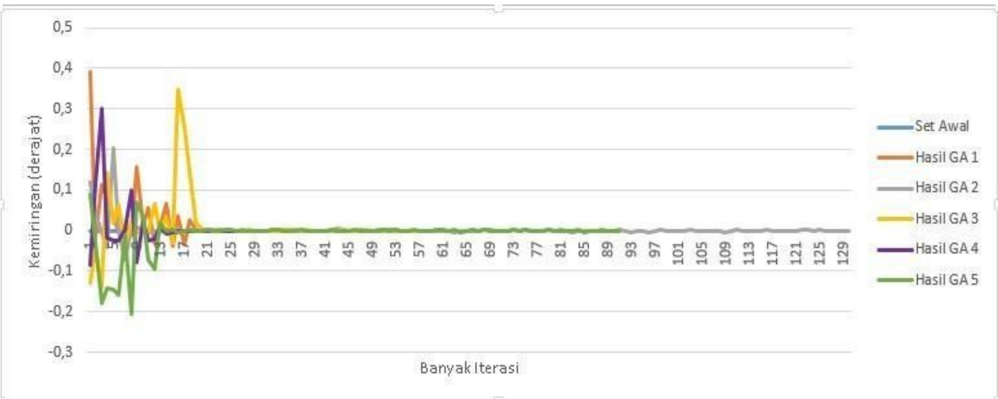

Fig. 5. Graph of genetic algorithm results on 0°

- Testing on 15°

Table 3. The result of the genetic algorithm on -15°

Lots of Trials	Many iterations	Initial Sets(°)	Error (°)	Errors on robots (°)	Time(s)
1	168	-15	0,00000236	1	62s
2	28	-15	0,00000419	0	12 s
3	25	-15	0,00000375	1	10 s
4	119	-15	0,00000797	0	42 s
5	47	-15	0,00000255	0	18 s

- Graph of genetic algorithm results at -15°

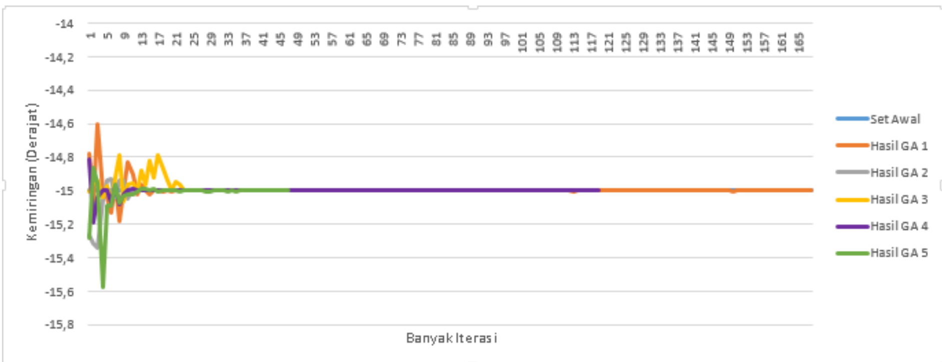

Fig. 6. The result of the genetic algorithm on -15°

- Testing on 17°

Table 4. The result of the genetic algorithm on 15°

The result of the genetic algorithm on 17°					
Lots of Trials	**Many iterations**	**Initial Sets(°)**	**Error (°)**	**Errors on robots (°)**	**Time(s)**
1	40	15	0,00000174	0	16.6 s
2	41	15	0,00000774	1	12.7 s
3	105	15	0,00000642	1	43 s
4	245	15	0,00000856	1	92.4 s
5	57	15	0,00000005	1	23.4s

- Graph of genetic algorithm results at 15°

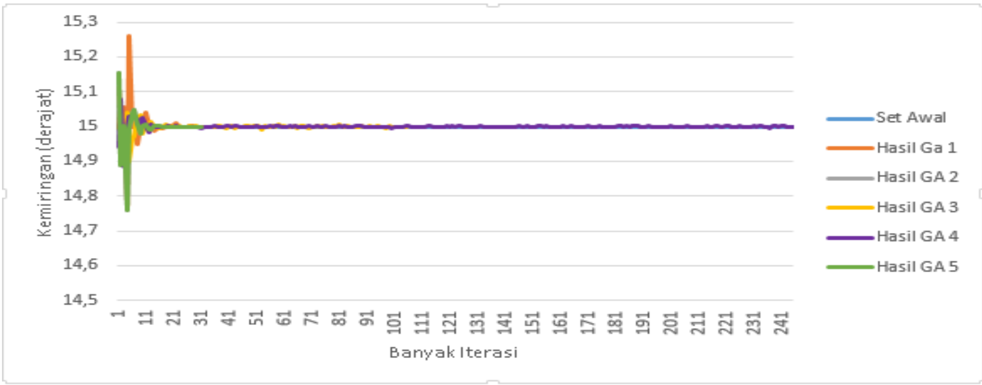

Fig. 7. The result of the genetic algorithm on 15°

- Testing on -17°

Table 5. The result of the genetic algorithm on -17°

The result of the genetic algorithm on -17°					
Lots of Trials	**Many iterations**	**Initial Sets(°)**	**Error (°)**	**Errors on robots (°)**	**Time(s)**
1	66	-17	0,000225	2	25.4s
2	223	-17	0,00000724	0	93 s

3	147	-17	0,00000932	1	61 s
4	265	-17	0,00000388	1	97.4 s
5	38	-17	0,0000089	-2	16 s

- Graph of genetic algorithm results at -17°

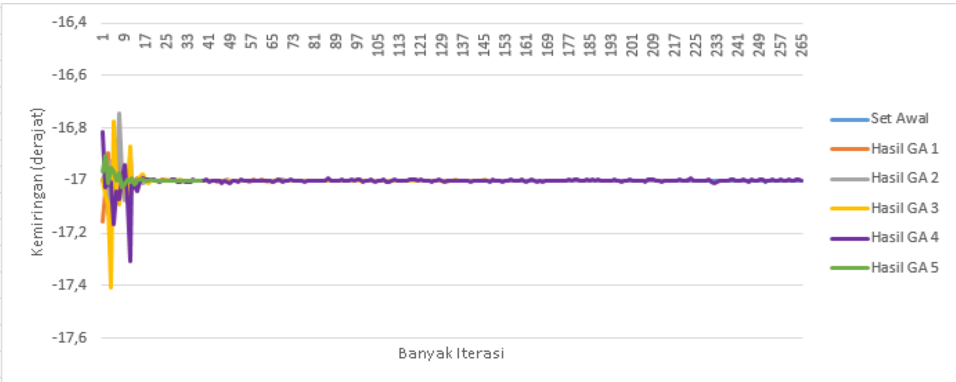

Fig. 8. The result of the genetic algorithm on -17°

- Testing on 17°

Table 6. The result of the genetic algorithm on 17°

The result of the genetic algorithm on 17°					
Lots ofTrials	**Many iterations**	**ial Sets(°)**	**Error (°)**	**Errors on robots (°)**	**Time(s)**
1	112	17	0,00000548	1	45 s
2	142	17	0,0000015	0	53 s
3	42	17	0,00000401	1	16 s
4	56	17	0,00000921	2	20 s
5	120	17	0,00000408	2	50.6 s

- Graph of genetic algorithm results at 17°

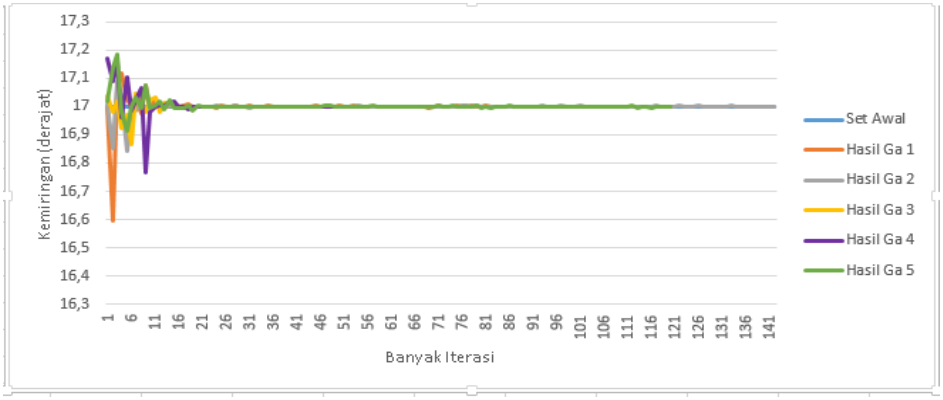

Fig. 9. The result of the genetic algorithm on 17°

Based on the results of this study, the errors resulting from the optimization of the geneticalgorithm are:

- At an angle of 0° with five trials, an average error is 0.00025 with a total time requiredof 2 minutes and 3 seconds.
- At an angle of -15° with five trials, an average error is 0.0000041 with a total timeneeded of 2 minutes and 4 seconds.
- At an angle of 15° with five trials, an average error is 0.0000049 with a total time of3 minutes 1 second.
- At an angle of -17° with five trials, an average error is 0.000056 with a total timerequired of 4 minutes and 8 seconds.
- At an angle of 17° with five trials, an average error of 0.0000048 is obtained with atotal time required of 3 minutes.

4. Conclusion

From this research, genetic algorithm method can be a solution for Barelang F.1 robot to overcome the obstacle cause it has a small error for the robot to stabilize itself, but this method is not recommended to be used directly in competitions because the iteration process takes a long time. It is better if the genetic algorithm is used on the Barelang F.1 robot to collect data of the obstacle for the competition.

Factors that affect the error on the robot can be caused by mechanical calculations of the feet that are not good enough with the calculation of the movement. Another factor that also affects the iteration time is the ability of the device that we used, if we used a device that has more high performance it might speed up the iteration process.

References

[1] N. E. Khomariah and S. Huda, "Udiebot: Implementasi Inverse Kinematic pada Quadruped Robot," *J. Ilmu Komput. dan Desain Komun. Vis.*, vol. 3, no. 1, pp. 12–21, 2018.

[2] A. I. Nasrudin, K. Anam, and M. A. Prawira N, "Evaluasi Inverse Kinematics untuk no.3, 2020, doi: 10.17529/jre.v15i3.14079.

[3] F. Z. Rachman and N. Yanti, "Robot Penjejak Ruangan Dengan Sensor Ultrasonik," *Jtt*,vol. 4, no. 2, pp. 114–119, 2016, [Online]. doi://doi.org/10.32487/jtt.v4i2.173

[4] Wahyu Tri Wibowo, "Implementasi Inverse Kinematics Pada Pergerakan Robot Quadruped," *Its*, pp. 1–110, 2017.

[5] D. Y. Habibi, "Penerapan Inverse Kinematic Pada Pengendalian Gerak Robot," 2011.

[6] R. C. Prayogo, A. Triwiyatno, and S. 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, doi://doi.org/10.14710/transient.v7i2.543-551

[7] I. A. Kurniawan, Feriyonika, and S. Pramono, "Inverse dan Body Kinematics pada Robot Hexapod," *Ind. Res. Work. Natl. Semin.*, vol. 9, pp. 115–123, 2018.

[8] M. Mudarris and S. G. Zain, "Implementasi Sensor Inertial Meansurenment Unit (IMU)untuk Monitoring Perilaku Roket," *Avitec*, vol. 2, no. 1, pp. 55–64, 2020, doi: 10.28989/avitec.v2i1.610.

[9] M. A. Akbar and A. Hidayatno, "Perancangan Perangkat Lunak Pendeteksi Sudut Dan Posisi Menggunakan Mikrokontroler Atmega 32," *Transmisi*, vol. 16, no. 1, pp. 49–53, 2014. doi://doi.org/10.12777/transmisi.16.1.49-53

[10] Eva Desiana, "Performance Algoritma Genetika (Ga) Pada Penjadwalan Mata Pelajaran," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 1, no. 1, pp. 56–60, 2016, doi: 10.30743/infotekjar.v1i1.42.

[11] S. F. Pane, R. Maulana Awangga, E. V. Rahcmadani, and S. Permana, "Implementasi Algoritma Genetika Untuk Optimalisasi Pelayanan Kependudukan," *J. Tekno Insentif*, vol. 13, no. 2, pp. 36–43, 2019, doi: 10.36787/jti.v13i2.130.
[12] A. Widhiyasa, "Kajian Genetic Algorithm," *Darwin*, pp. 1–11.

[12] [13] H. A. Sitompul, Implementasi Algoritma Genetika Untuk Menentukan Solusi Pada Optimisasi Nonlinier. Jurnal Darma Agung," vol. XXI, no. 9, pp. 183–188,