ATLANTIS PRESS

Atlantis Highlights in Engineering (AHE), volume 1

International Conference on Science and Technology (ICST 2018)

Designing and Implementing Trajectory Planning and Inverse Kinematics Algorithms using Hexapod Robot Platform

Miftachul Ulum[1], M. Khoirut[2], Haryanto[3], Riza Alfita[4], Diana Rahmawati[5], Kunto Aji Wibisono[6], Achmad Fiqhi Ibadillah[7]

Faculty of Engineering

University of Trunojoyo Madura

Bangkalan, Indonesia

[1]miftachul.ulum@trunojoyo.ac.id, [2]mkt.meemee@gmail.com, [3]haryanto_utm@yahoo.com, [4]riza.alfita@trunojoyo.ac.id, [5]diana_rahmawti@yahoo.com, [6]kunto.elektro@gmail.com, [7]fiqhi.achmad@gmail.com

***Abstract*—Among various forms of developed robots, legged robots are often used to overcome problems that cannot be solved by wheeled robots. The often occurred problem while developing this typical robot is about the flexibility and efficiency of the robot movement. In addressing the problem, this research aimed to design and implement trajectory planning and inverse kinematics algorithms using hexapod robot platform with three degrees of freedom on each leg. The use of both algorithms' models was to make the legged robots move smoother and easier to manage so that the robots would move according to the desired inputs. This research used smartphones as a support control system. By having 13 testing times with different angles, a perfectly moving robot corresponding to an input angle was amounted 31%, 54% for a moving robot with an offset angle below five degrees, and 15% for a robot moving with an offset angle above five degrees.**

***Keywords*—hexapod; inverse kinematics; trajectory planning**

I. Introduction

Robotics technology currently confronts a very rapid development, which is very influential and useful in several fields such as medicine, education, industry, agriculture, health, household, and assistive devices. In education scope, robots are used as research and development materials to produce something better and useful. Among various forms of developed robots, one of which is a mobile robot that can be classified into two types namely wheeled and legged robots. Wheeled robots are those that are able to maneuver using wheels, either with two or more wheels. Meanwhile, legged robots are those that can maneuver with artificial legs, such as, robots with 2 feet called biped robots, three-legged (tripod), four-legged (tetrapod), and six-legged robots (hexapod) [1].

Most wheeled robots' movement hardly experiences problems in its settings except for the two-wheeled types. On the contrary, the regulatory problems often arise in legged robots regardless the numbers of their legs. Legged robots consist of servo motors controlled in an angle to achieve the desired position. Such desired position can be determined by using a trial-error method that scholars often call it as forward kinematics. Unfortunately, such method remains inflexible and problematic. Therefore, this research aims to apply the inverse kinematics calculation method to automatically assign angle values to each servo motor and also to the trajectory planning system to make the robot's trajectory move more flexible. This research uses six legs or Hexapod robot with 3 degree of freedom (DOF) on each leg [2].

II. Review of Literatures

A. Hexapod Robot

Hexapod robot is a type of legged robots that moves using six feet or legs. This typical robot is statistically more stable because of the presence of the six legs compared to other types of the similar class. If one leg of the hexapod robot does not work well, the robot still can run. Moreover, not only for the sake of the robot stability, the other legs might help the robot move freely to find new footing to step and walk [3].

B. Inverse Kinematics

Inverse Kinematics is an analytical method for transforming Cartesian space to a joint space. In accordance to the kinematics equation, it can be obtained the relationship between the concepts of a joint space geometry in a robot with the concept of coordinates that are commonly used to determine the position of an object. With the kinematics model, the programmer can determine the input reference configuration fed to each actuator that enables the robot perform simultaneous movements to reach the desired position.

Inverse kinematics equation can be formulated by applying trigonometry by looking at each joint in one movement direction. Fig.1. shows the robot feet modeling in three dimensions mode (3D) [4][5].

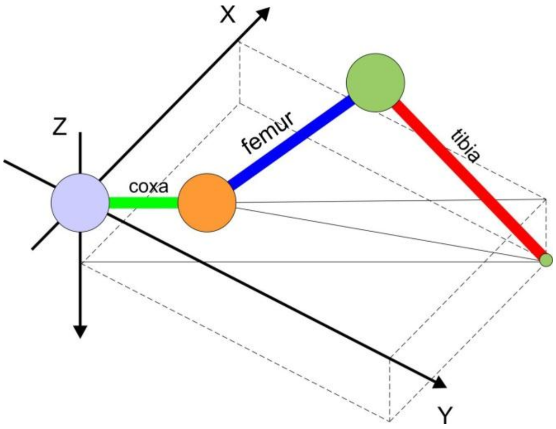

Fig. 1. Robot feet modelling in 3D mode

Simplifying by changing 3D to 2D modes is necessary to find the solution from the addressed inverse kinematics. To find the values of the joint parameter, it can be conducted by looking at the movement of the coxa link on the x and y axes in which the equation can be drawn as follows:

$$\alpha = \arctan\left(\frac{x}{v}\right) \quad (1)$$

In addition, Fig.2. shows the tangent function used in this research.

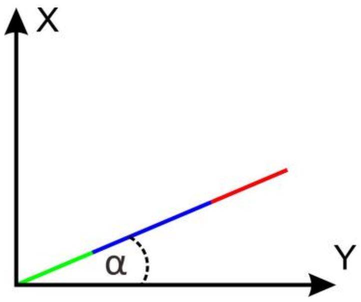

Fig. 2. Tangent function angles

The next vital parameter is the knee parameter value in which this parameter is important to make the femur move and to make the joint parameters move the tibia. Fig.3. portrays the 2-dimensional (2D) modeling of the robot legs [6].

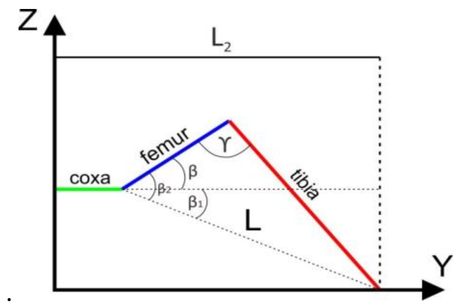

Fig. 3. Robot feet modelling in 2D modeL

$$L_2 = \sqrt{x^2 + y^2} \quad (2)$$

$$\beta_1 = \arctan\left(\frac{z}{L_2 - coxa}\right) \quad (3)$$

$$L = \sqrt{(L_2 - coxa)^2 + z^2} \quad (4)$$

By considering cosine rules, β and γ values can be obtained through the following equations:

$$\cos\beta_2 = \frac{L^2 + femur^2 - tibia^2}{2.L.femur} \quad (5)$$

$$\beta_2 = \arccos\left(\frac{L^2 + femur^2 - tibia^2}{2.L.femur}\right) \quad (6)$$

$$\beta = \beta_2 - \beta_1 \quad (7)$$

$$\cos\gamma = \frac{tibia^2 + femur^2 - L^2}{2.femur.tibia} \quad (8)$$

$$\gamma = \arccos\left(\frac{tibia^2 + femur^2 - L^2}{2.femur.tibia}\right) \quad (9)$$

In programming languages, there are several things to note covering:

a) The arc cos (acos) function does not have good accuracy in recovering cos values because cos(θ) = cos(-θ).

b) When sin (θ) is close to zero, i.e. θ° ≈ 0° or θ° ≈ 180°, a calculation error will be inaccurate and even not defined. In computer programming, such phenomena can cause the calculation to be NanN or indefinable. To prevent the occurrence of such calculation errors in programming, the calculation manipulation with the function used in the programming language should refer to atan2 (y, x) by looking at the quadrant of the AT2 function (arctan2).

$$\theta = \arctan 2(y, x) = \begin{cases} 0° \le \theta \le 90°, \text{for} + x \text{ and} + y \\ 90° \le \theta \le 180°, \text{for} - x \text{ and} + y \\ -180° \le \theta \le -90°, \text{for} - x \text{ and} - y \\ -90° \le \theta \le 0°, \text{for} + x \text{ and} - y \end{cases}$$

Couples of important notes regarding atan2(y,x) function include:

a) The atan function recovers the value between π/2 and -π/2

b) The atan2 function recovers the value between π and -π using x and y with numbers signed + or -.

c) $\tan\theta° = \dfrac{\sin\theta°}{\cos\theta°}$

Henceforth, the equation 5 can be changed into atan2 using a mathematical analogy of $\cos\beta_2 = A$ and $\sin\beta_2 = \pm\sqrt{1-A}$, so:

$$\beta_2 = \arctan 2\left(\frac{\sqrt{1-A^2}}{A}\right) \tag{10}$$

And, the equation 8 also can be changed using an analogy $\cos\gamma = D$ and $\sin\gamma = \pm\sqrt{1-D^2}$, so:

$$\gamma = \arctan 2\left(\frac{\sqrt{1-D^2}}{D}\right) \tag{11}$$

C. Trajectory Planning

Trajectory planning is a planning path or track used as a reference step on Hexapod robot. This trajectory planning uses the basic trigonometric concepts where the sin and cos functions are needed. Fig.4. shows a parabolic trajectory to make the feet able to move.

Fig. 4. Parabolic trajectory

However, this study only made four different trajectories that would be combined to finally create a parabolic trajectory in which the starting point lays on the middle (see Fig.5.). This is intended to no longer give transition effects on the robot when changing directions. So, the robot's initial and final positions will always be in the middle [7].

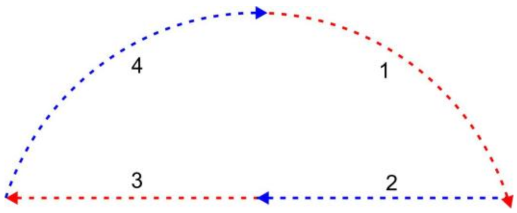

Fig. 5. Combined trajectories

To achieve the combined trajectories, the following equation must be encountered:

$$f_x = x\sin\left(\frac{t\times 90°}{s}\right) \tag{12}$$

$$f_y = y\sin\left(\frac{t\times 90°}{s}\right) \tag{13}$$

$$f_z = z - z\left(1-\sin\left(\frac{(s-t)\times 90°}{s}\right)\right) \tag{14}$$

x, y, and z are the displacement lengths for each axis. Whereas, t is the change in the displacement time and s is the amount of time changes for displacement. The greater the value of the time changes, the longer it will move.

The z value of tracks 2 and 3 is made to 0 so that the trajectory can be straight. Meanwhile, the t parameter will both increase and decrease in every track change and will be negative on tracks 3 and 4.

In this research, the robot runs using the three wave gait method, a walking method using 3 legs tread on the field and the other three legs lifting. When the legs tread on the field, the z values of tracks 1 and 4 are made into 0 while tracks 2 and 3 remain valuable according to the given input. Fig.6. helps visualize the track [8].

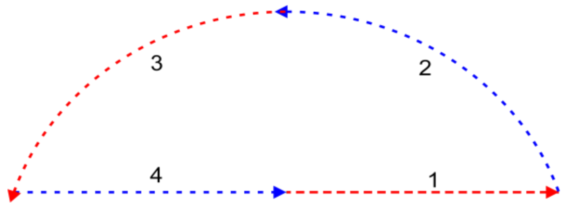

Fig. 6. Feet tracks when treading the ground

III. Results and Discussion

Several tests were carried out from the system that was made previously in three stages namely inverse kinematics test, trajectory planning test, and maneuvering test.

A. Inverse Kinematics Test

Inverse kinematics test was done by taking one-foot sample, then giving the input value to the robot in a form of end effector coordinates which further produced servo angles in accordance with the given input. The results of these angles were later confirmed by manual mathematical calculations.

In this experiment, coxa = 3 arm values, femur arm = 4.5, and tibia arm = 6.2. Whereas, the input coordinates were entered randomly but not exceeding the total length of the arm in each coordinate. Table 1 shows the results obtained from the experiment.

TABLE I. Results of Inverse Kinematics Test

Input Coordinates			**Calculation Results**			**Test Results**		
x	***y***	***z***	***θ₁***	***θ₂***	***θ₂***	***θ₁***	***θ₂***	***θ₃***
0	13.7	0	0	0	180	0	0	180
0	12	-1	0	44.81	114.69	0	45	114
0	11	2	0	33.81	99.6	0	34	99

Input Coordinates			Calculation Results			Test Results		
x	*y*	*z*	θ_1	θ_2	θ_2	θ_1	θ_2	θ_3
0	10	3	0	31.29	89.29	0	33	89
0	9	4	0	24.91	83.11	0	26	82
-5	7	2	**-35.54**	**51.62**	65.31	-36	52	65
-8	5	3	-57.99	34.99	81.45	-58	35	81
-6	6	4	-45	26.73	76.95	-45	27	77
-5	4	9	-51.34	-38.5	127.4	-52	-39	127
-8	5	7	-57.99	-14.96	124.63	-58	-15	125

The calculation value in Table I was obtained from the inverse kinematics calculation that had been explained previously. θ_1 value represented servo 1, θ_2 value represented servo2, and θ_3 value represented servo 3. The following was one proof of the calculation conducted to get the results shown in Table I. In this case the researchers took the 6th experiment sample namely x = -5, y = 7, and z = 2.

Further proof could be continued by referring to the results shown in Table II.

TABLE II. ERROR VALUES FOR EACH EXPERIMENT TIME

Input Coordinates			Errors		
x	*y*	*z*	θ_1	θ_2	θ_2
0	13.7	0	0	0	0
0	12	-1	0	0.19	0.69
0	11	2	0	0.19	0.6
0	10	3	0	1.71	0.29
0	9	4	0	1.09	1.11
-5	7	2	0.46	0.38	0.31
-8	5	3	0.01	0.01	0.45
-6	6	4	0	0.27	0.05
-5	4	9	0.66	0.5	0.4
-8	5	7	0.01	0.04	0.37
Total Errors			1.14	4.38	4.27
Average Errors			0.114	0.438	0.427
Average of the Total Errors			0.326		

In accordance with Table 1, the average error or offset of each servo angle was obtained from the difference between the calculation result and the test results (see Table 2).

From the test results of inverse kinematics, it could be seen that the error or offset was quite small, which was about 0.326 degrees. This result was good because it did not make too much changes to the servo angle.

B. *Trajectory Planning Test*

TABLE III. RESULTS OF TRAJECTORY PLANNING TEST

t	fx	fy	fz
0	0	0	2
1	0.618034	0	1.902113
2	1.175571	0	1.618034
3	1.618034	0	1.175571
4	1.902113	0	0.618034
5	2	0	0
4	1.902113	0	0
3	1.618034	0	0
2	1.175571	0	0
1	0.618034	0	0

t	fx	fy	fz
0	0	0	0
-1	-0.618034	0	0
-2	-1.175571	0	0
-3	-1.618034	0	0
-4	-1.902113	0	0
-5	-2	0	0
-4	-1.902113	0	0.618034
-3	-1.618034	0	1.175571
-2	-1.175571	0	1.618034
-1	-0.618034	0	1.902113
0	0	0	2

Trajectory planning test was also conducted by taking one foot sample. The robot foot was given a desired input in a form of the amount of time changes and the displacement lengths for each axis coordinate. Thus, the robot's legs could move according to the determined trajectory planning path or track.

In this test, the s value was 5 (the amount of displacement time), x length = 2, y length = 0, and z length = 2. In accordance with the trajectory planning formula described earlier, to make a parabolic trajectory, the merger between the trajectories was carried out and the results were shown in Table 3.

From the results depicted in Table III, it was then implemented on the robot's foot. In this case, programming each path using the "for" function was necessary and the robot foot would make a parabolic trajectory pattern like the graph shown in Fig. 7.

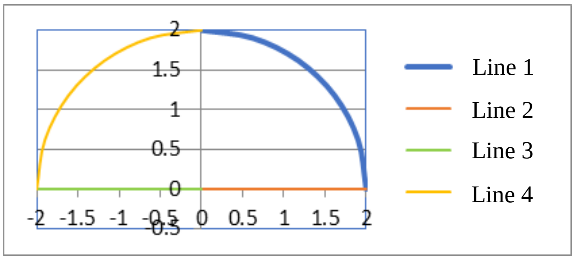

Fig. 7. Graphic results of trajectory planning test

C. *Maneuver Test*

In this test, the robot used a walking method with three wave gait. The three-wave gait method was a walking method using three feet to tread on the field while other three legs to move. The purpose of this test was to determine the efficiency and effectiveness of the running method of the three wave-gait combined with the use of inverse kinematics.

To get the x and y values, the researchers only needed to enter the length input values or the length of the steps and the desired angle of motion of the robot using the following formula.

$$x = lenght \times \cos\theta$$

$$y = lenght \times \sin\theta$$

During the test, the robot received data from the smartphone via Bluetooth and it would move according to the data received. The data received by the robot was in the form of rotation angle for maneuvering and determining range of steps and speed. The GUI interface on the smartphone could be seen in Fig.8.

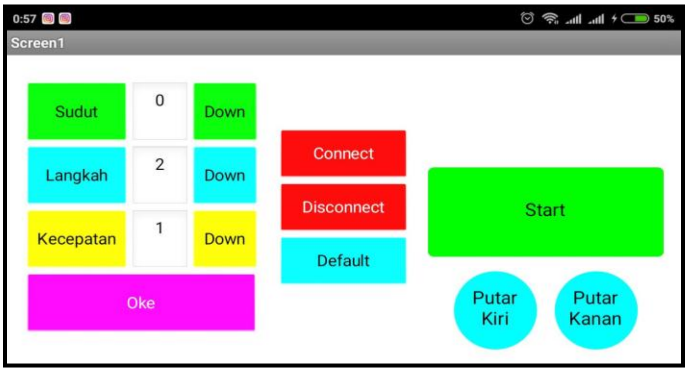

Fig. 8. GUI interface on smartphones

By sending data from a smartphone, the robot was able to pass the predetermined test track. Fig.9. shows the test track used in this test.

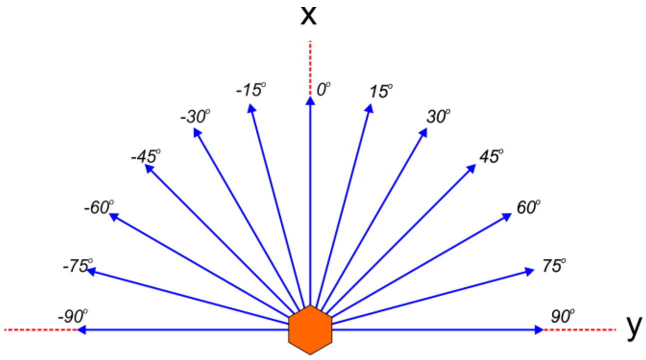

Fig. 9. Tracks in maneuvering test

TABLE IV. RESULTS OF MANEUVERING TEST

Determined Angle	Test Results	Offset
0	-1	1
15	22	7
30	31	1
45	43	2
60	62	2
75	75	0
90	90	0
-15	-13	2
-30	-30	0
-45	-41	4
-60	-66	6
-75	-75	0
-90	-92	2

For the 13 experiments with different angles, it could be seen that the robot moved perfectly through the trajectory four times (31%), traversed the trajectory at an offset below five degrees seven times (54%), and crossed the trajectory offset above five degrees twice (15%)

Based on the results found at the above tests, the movement of the robot was still not completely perfect caused by several factors including servo motors that had worn out and lacked of mechanical design and calibration of the inverse kinematics and trajectory planning on the robot.

IV. CONCLUSION

By referring to the test results and the implementation, this research concludes that the result between the undertaken inverse kinematics calculation and the real implementation shows error amounting 0.326. Meaning that, the result is good because the error level does not make a significant change in the servo angle. Moreover, the robot can actually move according to the desired trajectory using a trajectory planning, so that the movement of the robot is more effective. At last, this research identifies that, with the three wave gait walking method combined with the inverse kinematics and trajectory planning methods, the robot moves perfectly across the trajectory by 31%, crosses the trajectory with an offset below 5 degrees by 54%, and crosses the trajectory with an offset above 5 degrees by 15 %, in which the test is conducted for 13 times with different angles.

ACKNOWLEDGMENT

We would like to thank our collagues at Faculty of Engineering University of Trunojoyo Madura for all supports and reviewers of this conference.

REFERENCES

[1] A.D.S. Wibowo, Stability Of Walking Robot (Hexapod) With Inverse Kinematics, Bontang: Teknik Elektro Sekolah Tinggi Teknologi Bontang, 2015.

[2] E. Prasetya, Implementasi Inverse Kinematic Pada Pergerakan Mobile Robot Krpai Divisi Berkaki, Malang: Teknik Elektro Universitas Brawijaya. 2014.

[3] S. Setiawan, B.R. Firdaus, and Derisma, Penerapan Invers Kinematika Untuk Pergerakan Kaki Robot Biped, Padang: Teknik Elektro Politeknik Negeri Padang, 2015.

[4] Atique, M. Uddin, A.R. Ahad, Inverse Kinematics Solution for a 3DOF Robotic Structure using Denavit-Hartenberg Convention, Bangladesh: Department of Biomedical Physics & Technology University of Dhaka, 2015.

[5] A. Hidayat, "Desain Dan Implementasi Metode Inverse Kinematics Dan Sine Pattern Untuk Kontrol Gerak Pada Autonomous Quadruped Robot," unpublished.

[6] E. Abdy, "Kinematik Invers Untuk Pergerakan Kaki Robot Hexapod 3 Sendi," [Online], Available: https://wangready.wordpress.com/2012/06/27/kinematik-invers-untuk-pergerakan-kaki-robot-hexapod-3-sendi/. [Accessed: 8-Aug-2018].

[7] H.J. Hamilton, "Inverse Kinematics", [Online], Available: http://www2.cs.uregina.ca/~anima/408/Notes/Kinematics/InverseKinematics.htm. [Accessed: 8-Aug-2018].

[8] N. Liqing and H. Qingjiu, "Qingjiu, Inverse Kinematics for 6-DOF Manipulator by the Method of Sequential Retrieval," Adv. in Intell. Syst. Res., pp. 255-258, 2012 [Proceedings of the 1st International Conference on Mechanical Engineering and Material Science, 2012].