Journal of Physics:
Conference Series

PAPER • OPEN ACCESS

Walking Pattern for Quadruped as Observer Robot

To cite this article: Nuril Esti Khomariah and Samsul Huda 2019 *J. Phys.: Conf. Ser.* **1201** 012010

IOP Conf. Series: Journal of Physics: Conf. Series **1201** (2019) 012010 doi:10.1088/1742-6596/1201/1/012010

Walking Pattern for Quadruped as Observer Robot

Nuril Esti Khomariah[1] **and Samsul Huda**[2]

Department of Informatics Engineering, Universitas 17 Agustus 1945 Surabaya, Indonesia

E-mail : [1]nuril@untag-sby.ac.id; [2]samsul@untag-sby.ac.id

Abstract. This paper explains about the walking patterns of quadruped robots. This robot was made to be an observer robot. Therefore, being slick in movement is the main feature that must be owned by this robot. As a four-legged robot, the movement (forward, backward, shift left, shift right, rotate) requires an algorithm to control the synchronization movements of each leg. This algorithm consists of several poses (including the body, as well as all legs) and if arranged in sequence it will form a certain pattern of movement. Each leg has 3 DOF. The pose of each leg is represented in the cartesian coordinates (x, y, z axis), while the movement of each joint is represented in angular. It has solved using the inverse kinematics formula which is useful for converting leg coordinates into angular for each leg joint. By defining the walking pattern, the robot can more easily controlled by the direction and speed of its movement.

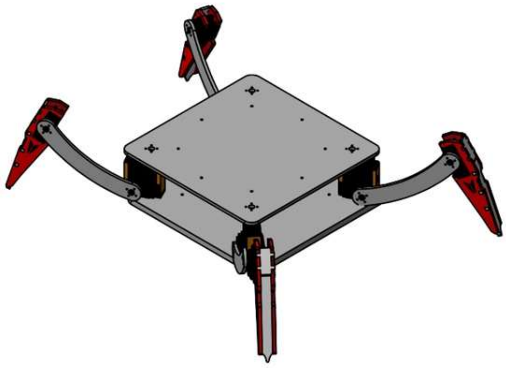
1. Introduction

Indonesia is one of the countries that has the most volcanoes in the world. This beauty certainly attracts many tourists from both local and international. With the increase in climbing activities, of course increasing the number of accidents. The most common occurrence is that climbers get lost, climbers fall into poisonous craters or enter caves with low oxygen levels. Large, hard-to-reach and dangerous areas are the main obstacles for rescue teams. While the faster the climber was found, the presentation of the possibility of the climber survived even greater. In this paper, proposing an observer robot is useful to help speed up the task of rescue teams in conducting searches for victims in mountainous areas[1].

Lately there have been more and more researches on robots, especially legged robots, such as two-legged (biped)[2], three-legged (tripod), four-legged (quadruped)[3], six-legged (hexapod)[4]. Mechanical design of multi-legged robot certainly has advantages and disadvantages compared to wheeled robots. Multi-legged robots are capable of better crossing uneven areas such as rocks [1], but not faster than wheeled robots. With a lot of foot configuration, of course, it will cause its own problems such as synchronizing the movements of each robot leg to the desired position and adjusting the pattern of movement [3] [4].

2. Overview of The *Platform* : UdieBot

UdieBot is a quadruped robot that has 3 DOF in each leg, as shown in Figure 1. Each DOF represents a joint owned by a robot. To be able to move the joint, an actuator is needed. In this robot the actuator used is a servo motor. So the total servo motor used on the whole robot leg same as number of joints is 12 servo motors. In order to get synchronous movements for the entire joint of the leg, the same type of servo motor is used (Hitec HS-7954SH).



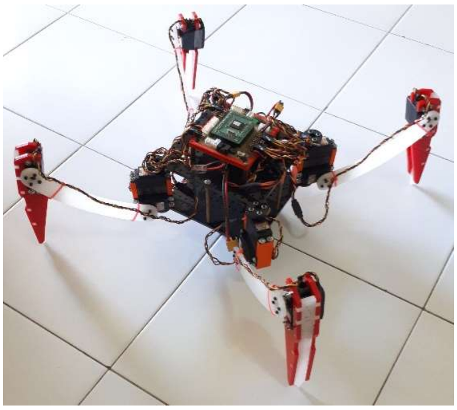


Figure 1. Quadruped as observer robot

The weight of this robot is 1.56 Kg with a height of 200 mm, a length of 350 mm and a width of 350 mm. The DOF total on a robot is 12 DOF with details 3 DOF for each leg. For other specifications such as weight, body dimensions, power supply and type of controller used can be seen in Table 1. There are 2 (two) controllers that both use STM32F4 with a speed of 168 MHz. The first controller is used to process the entire system including data communication, retrieval and processing sensor data. While the second controller is used to regulate robot movement / navigation.

Table 1. Robot specifications.

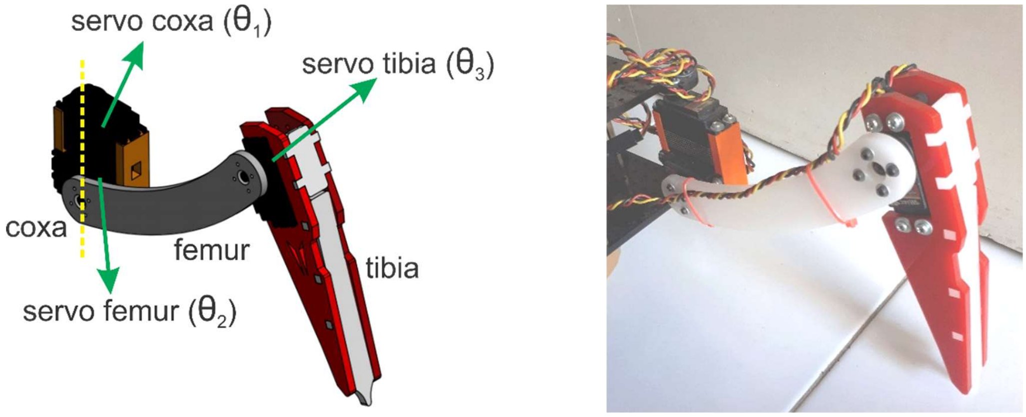
Specifications	Details
Type	Quadruped (four legged)
Weight	1.56 Kg
Height	200 mm
Length	350 mm
Width	350 mm
DOF	12 DOF
Actuators	12x servo motors (Hitec HS-7954SH)
Power Supply	11.1 Volt 5Ah
Controller	2x STM32F4 ARM (168MHz)

3. Mechanical Design

The first step when designing a robot is solved its mechanical. Udiebot's mechanical design is a four legged robot with 3 DOF on each leg, as shown in Figure 1. In order to controlling the motion of Udiebot such as going forward, backward, shift right, shift left or rotate, the kinematic modelling must be completed first. Therefore, the inverse kinematics equation is needed by completing the forward kinematics first. The details about mechanical design of the overall robot consisting of legs, forward kinematic and inverse kinematic already done before [5].

3.1. Legs

Udiebot's leg mechanism consist of 3 servo motors namely servo coxa (θ_1), servo femur (θ_2) and servo tibia (θ_3). It is has 12 DOF total, the detail is shown in Table 2. It was combine servo coxa and servo femur in one frame on hip. Whilst servo tibia was attached on knee, see Figure 2.



(a) The servo position on each leg (b) Side view

Figure 2. Robot's leg.

Almost all mechanical was made from 3D printer and the others from acrylic sheets. The part with black color shown the servos motors.The frames printed from 3D printer shown in orange part, while the others frames made from cutted acrylic. Table 2 and Table 3 shown the link length and position of each actuator for the Udiebot's leg.

Table 2. Leg parameters.

Parameters	Length (mm)
coxa	60
femur	150
tibia	150

Table 3. Actuators and DOF.

Joint		Actuator	DOF
hip	pitch	1x servo motor (femur)	1 DOF
	yaw	1x servo motor (coxa)	1 DOF
knee	pitch	1x servo motor (tibia)	1 DOF
Total		**4 legs x 3 DOF**	**12 DOF**

Udiebot's leg have two actuators on hip (pitch, yaw) which called servo motor femur and coxa. While, there is only one actuator on knee (pitch) which called servo motor tibia. Since the robot have 4 legs, so the robot have 12 DOF total.

3.2. Forward Kinematics

Forward kinematics is used to obtain kinematic equations that produce the position coordinates and the orientation of the robot EoF (End of Effector) for each joint / joint. Denavit-Hartenberg (DH) Parameters used to state for each homogeneous matrix.

$$H_3^0 = H_1^0 H_2^1 H_3^2 \qquad (1)$$

$$H_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & \alpha_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & \alpha_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

3.3. Inverse Kinematics

In general, inverse kinematic is used to move a manipulator robot based on the position of the desired coordinate point. To be able to move the robot's legs we must define the position of the coordinates in the Cartesian space first. This process requires input data in the form of coordinate points (x, y, z) that produce output data in the form of angular values for each robot leg joint.

$$legLength = \sqrt{posX^2 + posY^2} \tag{3}$$

$$Zlength = \sqrt{posZ^2 + legLength^2} \tag{4}$$

$$\theta_A = \tan^{-1}((legLength - coxaLength)/posY) \tag{5}$$

$$J = \cos^{-1}\left(\frac{Zlength^2 - femurLength^2 - tibiaLength^2}{-2.FemurLength\,.Zlength}\right) \tag{6}$$

$$K = \cos^{-1}\left(\frac{femurLength^2 + tibiaLength^2 - Zlength^2}{-2.FemurLength\,.tibiaLength}\right) \tag{7}$$

$$\theta_1 = \tan^{-1}(posX, posY) \tag{8}$$

$$\theta_2 = J + \theta_A \tag{9}$$

$$\theta_3 = 90^o - (K) \tag{10}$$

Equation 3-7 is used to find the value of supporting parameters. While Equation 8-10 is the main formula that calculate final value for θ_1, θ_2 and θ_3.

4. Walking Pattern

In order to controlled move, it is necessary to define the walking pattern which actually consists of several robot poses. In one period of walking, there are 5 robot poses needed, as shown in Figure 3. The robot's leg are moved one by one alternately in order to remain balanced by paying attention to the CoM point (center of mass).

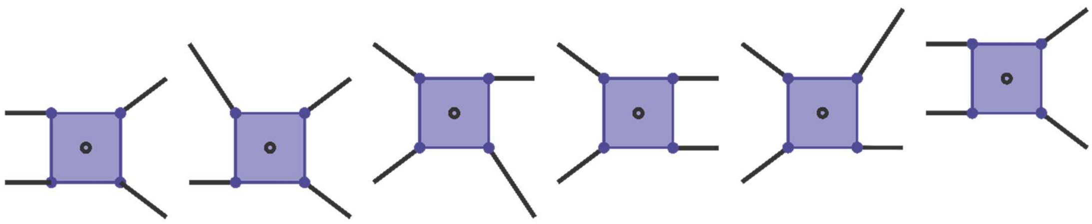


Figure 3. Pose of walking pattern in one periode.

While in Figure 4, it explains the step by step transfer of poses to other poses in first half periode walking. The square at toe illustrates that the leg touch the floor. While the dashing line illustrates that the leg being lifted. Followed by Figure 5 where it explains the step by step transfer of poses to other poses in second half periode walking.

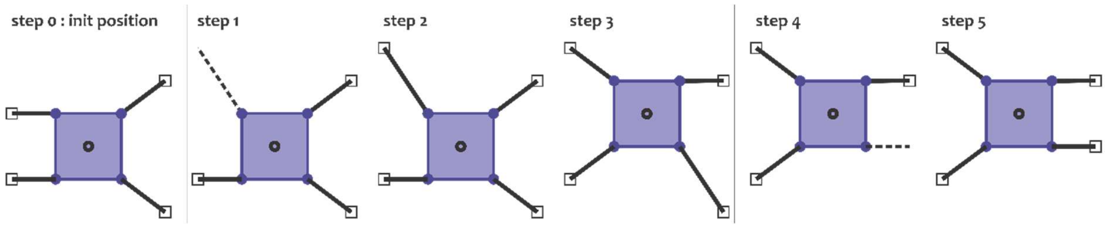


Figure 4. First half periode.

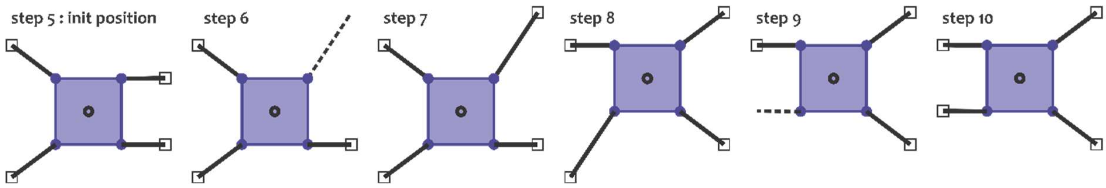


Figure 5. Second half periode.

The balance criteria is maintain the position of CoM (center of mass) robot always inside the SP (support polygon). The position of CoM robot can be calculate using Equation 11-13.

$$x_{com} = \frac{\sum_{i=1}^{n}(x_i mass_i)}{mass_{total}} \tag{11}$$

$$y_{com} = \frac{\sum_{i=1}^{n}(y_i mass_i)}{mass_{total}} \tag{12}$$

$$z_{com} = \frac{\sum_{i=1}^{n}(z_i mass_i)}{mass_{total}} \tag{13}$$

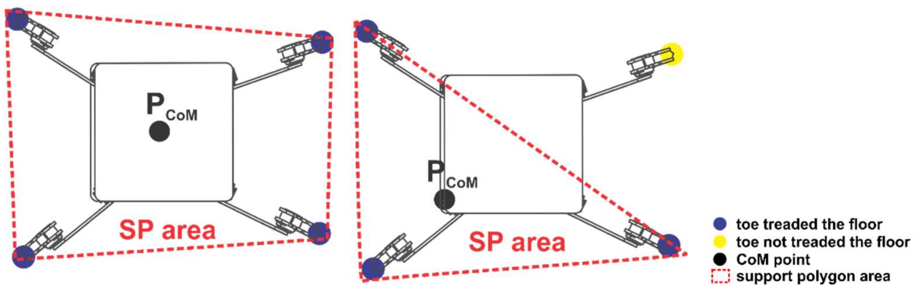


Figure 6. Center of mass point.

Then the CoM must be projected into floor which is the value of z-axis defined by zero. After that, if the CoM position still inside the Support Polygon area, then the robot is in balance condition. Figure 6 shows the position of CoM when 4 toe treaded the floor. When only 3 toe treaded the floor, the CoM position was shifted to the bottom left from original position but still inside the Support Polygon area.

IOP Conf. Series: Journal of Physics: Conf. Series **1201** (2019) 012010 doi:10.1088/1742-6596/1201/1/012010

5. Experimental Result

Walking pattern in the section above implemented by using UdieBot robot. This section discuss about the experimental result of walking pattern by analyzing the displacement of robot's position and its walking speed. Target coordinate, actual coordinate and error displacement shows in Table 4. It is shows that the displacement error got bigger when the distance displacement father away.

Table 4. Displacement error X and Y axis.

Target Coordinate (cm)		Actual Coordinate (cm)		Error (cm)		Diagonal Distance (cm)	Time (sec)	Walking Speed (m/s)
X	Y	X	Y	X	Y			
10	0	10	0	0.00	0.00	10.00	4.02	0.025
20	0	20.3	0.4	0.30	0.40	20.30	8.17	0.025
30	0	30.5	0.8	0.50	0.80	30.51	12.32	0.025
40	0	40.6	1.1	0.60	1.10	40.61	16.44	0.025
50	0	50.9	1.3	0.90	1.30	50.92	20.50	0.025
60	0	60.9	1.6	0.90	1.60	60.92	24.83	0.025
70	0	71.1	2	1.10	2.00	71.13	29.21	0.024
80	0	82.3	2.2	2.30	2.20	82.33	33.76	0.024
90	0	93.6	2.6	3.60	2.60	93.64	38.66	0.024
100	0	104.9	3.1	4.90	3.10	104.95	43.22	0.024

Table 5. Speed and Displacement.

Time (sec)	Walking Speed (m/s)	Target Coordinate (cm)		Actual Coordinate (cm)		Error (cm)	
		X	Y	X	Y	X	Y
9.23	0.108	100	0	110.5	9.8	10.50	9.80
18.85	0.053	100	0	109.3	8.1	9.30	8.10
28.18	0.035	100	0	107.6	6.7	7.60	6.70
36.89	0.027	100	0	105.2	5.1	5.20	5.10
46.00	0.022	100	0	104.6	4.3	4.60	4.30
55.63	0.018	100	0	103.9	3.6	3.90	3.60
65.04	0.015	100	0	103.3	3.1	3.30	3.10
75.16	0.013	100	0	102.6	2.5	2.60	2.50
84.37	0.012	100	0	101.8	2.1	1.80	2.10
94.42	0.011	100	0	101.1	1.7	1.10	1.70

The next experiment is compared displacement errors based on differences walking speed. Table 5 shows that robot is made walking along 100 cm with 10 different speeds. The result is when increase the walking speed also impact on increasing displacement error in x and y axis.

6. Conclusion

This paper proposed a walking pattern for quadruped robot. The walking pattern consist of several poses that still consider the CoM point for stable condition. The pattern generated fixed under speed and distance of step walking. The experimental result show that when the walking speed is increase, then the displacement error gets bigger.

7. Future Work

In this paper explain about walking patterns that are still static. In the future it will be further developed with additional stability control so that the robot can easily maintain its balanced position in uneven terrain. And finally, additional distance detection sensor is needed so the robot can automatically avoid the obstacle which is blocking the path of the robot.

Acknowledgments

This work was supported by Universitas 17 Agustus 1945 Surabaya and The Ministry of Research, Technology and Higher Education of Indonesia (DRPM Kemenristek Dikti) through Penelitian Dosen Pemula (PDP) research fund.

References

[1] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "BigDog, the Rough-Terrain Quadruped Robot," in *The International Federation of Automatic Control*, 2008, pp. 10822–10825.

[2] N. E. Khomariah, D. Pramadihanto, and R. S. Dewanto, "FLoW bipedal robot: Walking pattern generation," in *Proceedings - 2015 International Electronics Symposium: Emerging Technology in Electronic and Information, IES 2015*, 2016.

[3] J. Li, J. Wang, S. X. Yang, K. Zhou, and H. Tang, "Gait planning and stability control of a quadruped robot," *Comput. Intell. Neurosci.*, vol. 2016, pp. 335–341, 2016.

[4] Y. Ya-xin and J. Bo, "Gait Generation and Control of a Hexapod Walking Robot," *Open Electr. Electron. Eng. J.*, vol. 8, no. 1, pp. 335–341, 2014.

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