Autonomous Quadruped Robot Locomotion Control Using Inverse Kinematics and Sine Pattern Methods

Aditya Hidayat
School of Electrical Engineering
Telkom University
Bandung, Indonesia
adityahidayat@students.telkomuniversity.ac.id

Agung Nugroho Jati
School of Electrical Engineering
Telkom University
Bandung, Indonesia
agungnj@telkomuniversity.ac.id

Randy Erfa Saputra
School of Electrical Engineering
Telkom University
Bandung, Indonesia
resaputra@telkomuniversity.ac.id

***Abstract*—In this paper, it will be explained how to solve robot locomotion using inverse kinematics and sine pattern methods on autonomous quadruped robot with 3 DOF for each leg using geometrical approach. It also includes python code examples inside. By using inverse kinematic method, it could be found the angle for every actuator from end effector position by calculation within the method of sine pattern which can generates values group of end effector position in a pattern of sine wave. So that it could solve multi value problem on inverse kinematics.**

Keywords—locomotion control; inverse kinematics; sine pattern; quadruped robot.

I. Introduction

Quadruped robot or legged robot is more adaptive to terrain than wheeled robot in terms of land exploration, however it also has its own problem, and the main problem of quadruped robot is its locomotive calculation that can be more complex than wheeled robot, therefore we need to choose between inverse kinematics or forward kinematics as the method for the locomotion of the robot, and we have to find the right calculation for each method.

Forward kinematics and inverse kinematics is two general locomotive methods that can be used in legged robot. These two methods are opposite to each other. Forward kinematics is focused on the process that we can find end effector position from angles values at each actuators, while inverse kinematics is focused on the result, that we can find angles values at each actuators from known end effector position, which is easier to implement on legged robot with many legs and DOFs.

Sine pattern is a method that can be used together with inverse kinematics to simplify the locomotion variable, it will generate values group of end effector position in a sine wave pattern [1] that can be used on inverse kinematics calculation.

This paper is focused on how to solve inverse kinematics on quadruped robot that has 3 DOF on each leg with geometrical approach and how to simplify the variable of calculation using the sine pattern, this paper also focused on the pseudo code program calculation rather than the theoretical calculation.

This paper is structured as follows. First, in section II, a description of other related work is provided. Next, section III presents explanation and implementation of inverse kinematics and sine pattern methods. Then, in section IV, testing and analysis as result of proposed method implementation is described. Finally, in section V, some conclusions and future research are explained.

II. Related Work

Yamazaki et al. [2] explained the design and control on SCOUT, a simple quadruped robot with 1 DOF. They presented hardware explanation, and also explain the locomotion on SCOUT robot with many figures and detail explanation for each section.

Meng et al. [3] focused on kinematics calculation on quadruped robot. Both inverse and forward kinematics were explained on this paper with complete formula for each method.

Morse et al. [4] implemented new method for quadruped robot locomotion using SUPG (single unit pattern generator) as solution for CTRNNs or sinusoidal pattern. The aim of this method is to obtain better long term stability than other existing pattern generator method.

Wang [5] explained comparison between gaits based on mathematical model, and also implemented on quadruped robot with 3 DOF on each robot's leg. He also used central pattern generators on its robot's implementation.

III. Proposed Method

To initiate the required calculation, we need to know the parameters associated with the system as shown in the Figure 1 below that shows one leg system geometry with 3 DOF, and it shows all the parameter we need in x,y,z axis, r for the length of legs, *p(x,y,z)* for the end effector position, θ for the joint angles, and α for the reach distance of the legs.

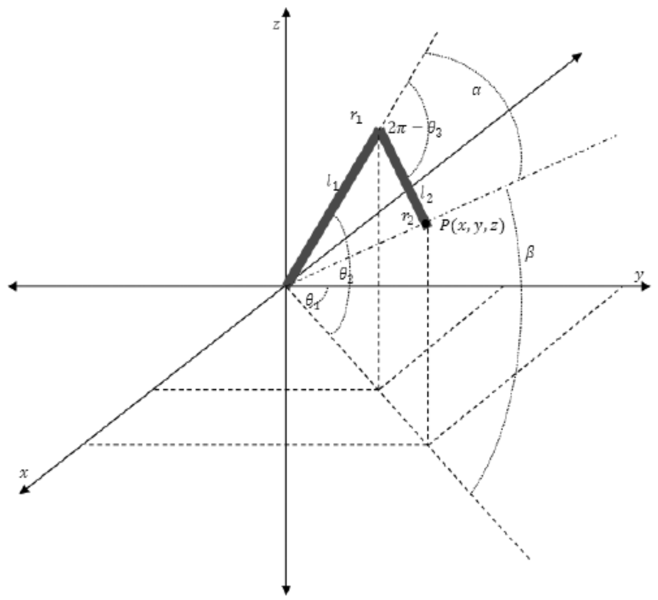

Fig. 1. Leg system geometry
[edited from WR note Electrical Engineering]

A. Inverse Kinematics

The purpose of inverse kinematics method is to find ($\theta_1, \theta_2, \theta_3$) and ($x,y,z$) as the input. The ($x,y,z$) value generating process will be described in sine pattern section.

From Figure 1 we have:

$$x = l1\sin\theta_1\cos\theta_2 + l2\sin\theta_1\cos\theta_2\cos\theta_3 - l2\sin\theta_1\sin\theta_2\sin\theta_3 \quad (1)$$

$$y = l1\sin\theta_1\cos\theta_2 + l2\cos\theta_1\cos\theta_2\cos\theta_3 - l2\cos\theta_1\sin\theta_2\sin\theta_3 \quad (2)$$

$$z = l1\sin\theta_2 + l2\sin\theta_2\cos\theta_3 + l2\cos\theta_2\sin\theta_3 \quad (3)$$

We can find $\theta_1, \theta_2, \theta_3$:

$$\theta_1 = \tan^{-1}\left[\frac{x}{y}\right] \quad (4)$$

$$\theta_2 = \tan^{-1}\left[\frac{z(l_2\cos\theta_3 + l_1) - \left(\sqrt{x^2+y^2}\right)(l_2\sin\theta_3)}{2l_1l_2}\right] \quad (5)$$

$$\theta_3 = \cos^{-1}\left[\frac{x^2+y^2+z^2-l_1^2-l_2^2}{2l_1l_2}\right] \quad (6)$$

However, we still have a problem using the (5) and (6) formula, because it would produce inaccurate result due to the following reasons:

1. $\cos(\theta)=\cos(-\theta)$, so that $\cos^{-1}$ will return bad result for *cos* value
2. when $\sin(\theta)$ has $\theta\approx0^o$ and $\theta\approx180^o$ on the formula, it will return an inaccurate result due to 0 value.

Better result of the degree values can be obtained with convert the (5) and (6) to atan2, because it will return values of π and $-\pi$ with signed x and y values [6]. The final result for inverse kinematics calculation would be like this:

$$\theta_1 = \text{atan2}(x, y) \quad (7)$$

$$\theta_2 = \text{atan2}(\pm C, D) \quad (8)$$

$$\theta_3 = \text{atan2}(\pm\sqrt{1-A^2}, A) \quad (9)$$

$\theta_1, \theta_2, \theta_3$ can be used on further calculation after transform its value from radian to degree form depend on what kind of further calculation we need(e.g. add with actuator's set point).

This is the example code for inverse kinematics according to (7), (8), and (9) formula:

```
....

    F=((x*x)+(y*y)+(z*z)-(l1*l1)-
(l2*l2))
    E= (2*l1*l2)
    A=F/E
    B=math.sqrt(math.fabs(1-(A*A)))
    tetha3=math.atan2(-B,A)
    buff_1=(l2*math.cos(tetha3))+l1
    buff_2=math.sqrt((x*x)+(y*y))
    buff_3=l2*math.sin(tetha3)
    C=(z*buff_1)-(buff_2*buff_3)
    D=(buff_1*buff_2)+(z*buff_3)
    tetha2=math.atan2(math.fabs(C),D)
    tetha1=math.atan2(x,y)
```

B. Sine Pattern

Problem that may occur on inverse kinematics is how to determine the end effector position as a group values for its calculation. Therefore, we can use sine pattern to solve those problem. Sine Pattern is a method that can simulate the pattern of end effector position as a sinusoidal wave pattern, for locomotive problem, the 1st actuator is active on sinusoidal pattern and the 2nd and 3rd actuator only active on positive or negative sine pattern, as shown on figure 2.

978-1-5090-0744-8/16/$31.00 ©2016 IEEE

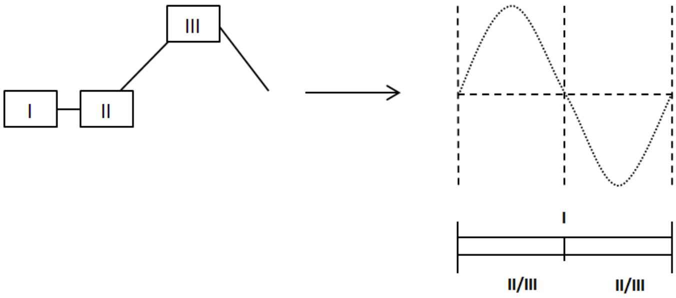

Fig. 2. Sine pattern based on each actuators movement phase

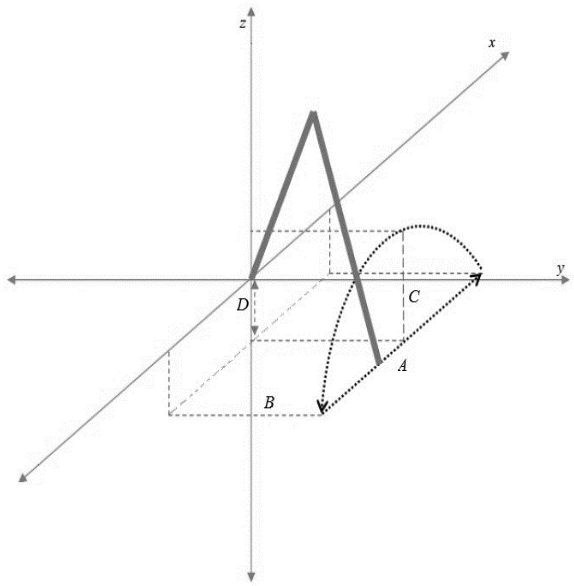

Fig.3. Sine pattern variables
[edited from WR note Electrical Engineering]

Sine pattern calculation purpose is to find the end effector position *(x,y,z)* in sine wave pattern. First, we have to understand the variable used on sine pattern as show on figure 3. However we still have to determine input variables value *(A, B, C, D, α)* to find desired leg's movement we need.

The variety of sine pattern formula is depending on phase of the leg's movement and direction of the leg's movement variable. Each of those variables has values of 0 and 1, which represent movement of the leg. The formula of each *(x,y,z)* is vary, depend on the movement of the leg.

On the sine pattern code example, each angle values are multiplying by 0.0174532925 to directly convert the angle value into radian. The angle variable on the code is come from incremental angle value for continuous locomotion purpose on robot's main program, which is the incremental value is determining smoothness of robot's locomotion. The smaller value will generate smoother locomotion.

TABLE I. FORMULA TABLE FOR (X,Y) VARIABLE

Direction	Phase	x	y
0	0	$x=-\frac{1}{2}A\cos(90^\circ-\alpha)\sin\omega$	$y=-\frac{1}{2}A\sin(90^\circ-\alpha)\sin\omega+B$
0	1	$x=\frac{1}{2}A\cos(90^\circ-\alpha)\sin\omega$	$y=-\frac{1}{2}A\sin(90^\circ-\alpha)\sin\omega+B$
1	0	$x=-\frac{1}{2}-A\cos(90^\circ-\alpha)\sin\omega$	$y=-\frac{1}{2}-A\sin(90^\circ-\alpha)\sin\omega+B$
1	1	$x=\frac{1}{2}-A\cos(90^\circ-\alpha)\sin\omega$	$y=-\frac{1}{2}-A\sin(90^\circ-\alpha)\sin\omega+B$

TABLE II. FORMULA TABLE FOR (Z) VARIABLE

Direksi	Fasa	z
0/1	0	$z=\begin{cases}-D & ,0^\circ\le\omega<90^\circ \\ C\sin(\omega-90^\circ)-D & ,90^\circ\le\omega<270^\circ \\ -D & ,270^\circ\le\omega<360^\circ\end{cases}$
	1	$z=\begin{cases}-C\sin(\omega-90^\circ)-D & ,0^\circ\le\omega<90^\circ \\ -D & ,90^\circ\le\omega<270^\circ \\ -C\sin(\omega-90^\circ)-D & ,270^\circ\le\omega<360^\circ\end{cases}$

Code examples for sine pattern calculation using direction = 0 :

```
if direksi == 0:
              var_y1 = ((-
0.5)*(A)*math.sin((90-
alpa)*0.0174532925)*math.sin(sudut*0.017
4532925)) + B
              if phasa == 0:
              var_x1 = (-
0.5)*(A)*math.cos((90-
alpa)*0.0174532925)*math.sin(sudut*0
.0174532925)
       if(sudut>=0) and (sudut<90):
              var_z1=-D
elif (sudut>=90) and (sudut<270):

       var_z1=((C*math.sin((sudut-
90)*0.0174532925))-D)
                            elif
(sudut>=270) and (sudut<360):
                     var_z1=-D
               elif phasa == 1:
                            var_x1 =
(0.5)*(A)*math.cos((90-
alpa)*0.0174532925)*math.sin(sudut*0.017
4532925)                            if
(sudut>=0) and (sudut<90):
                     var_z1=((-
C*math.sin((sudut-90)*0.0174532925))-D)
              elif (sudut>=90) and
(sudut<270):
       var_z1=-D
       elif (sudut>=270) and
(sudut<360):
       var_z1=(((-C)*math.sin((sudut-
90)*0.0174532925))-D)
```

C. Trot Gait

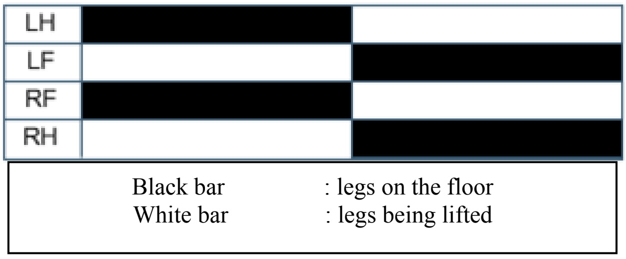

Fig. 3. Movement of trot gait

Sine pattern method has close relation with the walking gaits that is used by the robot, it is because phase variable used on sine pattern will determine the gait used on robot. The autonomous quadruped robot used in this research is using trot gait as its walking gaits. Trot gait is one of the walking gaits that commonly used on quadruped robot, the gaits itself is an adaptation from horse walking gait. The advantage of this gait is on its speed and maneuver aspects.

The pattern of this gait is using two legs in diagonal position to take a step, while the other two legs sustain the stability [5]. To understand the gait and the phase, the legs named by right front (RF), right hind (RH), left front (LF), and left hind (LH). With using trot gait, the RF leg phase value will be the same as LH and the phase of LF will be the same as the RH leg, as shown on figure 3.

D. Implementation

Implementation of inverse kinematics and sine pattern methods will be carried out on Autonomous Quadruped Robot with 3 DOF on each leg with using Beaglebone Black as its controller and motor servo Dynamixel AX-12A as its actuators. Figure 4 shows the current state of Autonomous Quadruped Robot.

Figure 5 is explained about the whole locomotion control on autonomous quadruped robot with following steps:

1. Get input for sine pattern calculation
2. Performing sine pattern calculation
3. Get sine pattern output and other value needed for input in inverse kinematics
4. Performing inverse kinematics calculation
5. Get θ_1, θ_2, θ_1 as output from inverse kinematics calculation
6. Add the value with servo set point
7. Performing incremental function for angle variable
8. If angle value less or equal to 360 it will update new angle variable, if not, the program will end.

Fig. 4. Current state of Autonomous Quadruped Robot

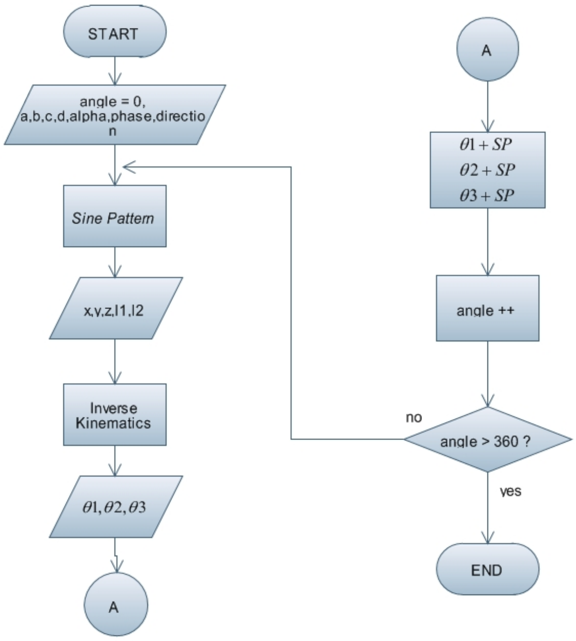

Fig. 5. Locomotion control flowchart

E. Issue

The main issue in this methods is some variables on sine pattern calculation are user defined variables, that means we still have to define the variables on our own with trial and error method to obtain the optimal or desired locomotion model we need, because it depends on robot's movement, body, and legs model, which can be different from one to other robots.

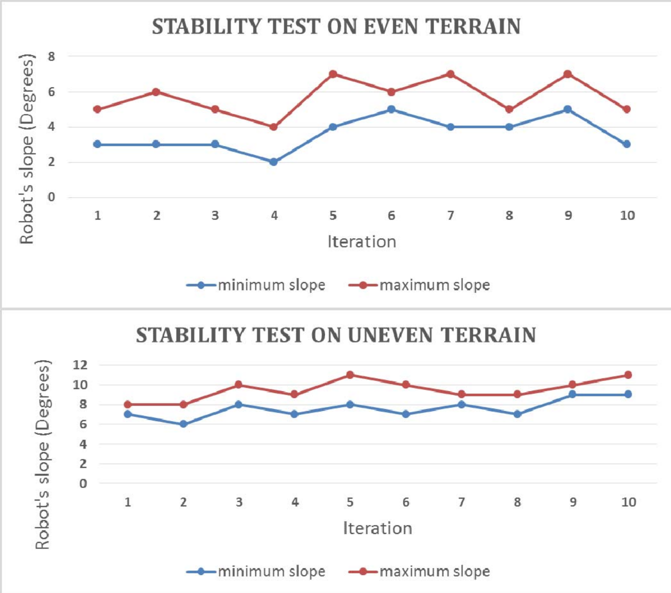

Fig. 6. Stability test graph

IV. RESULT

A. Testing

The aim of this test is to know locomotion stability using inverse kinematics and sine pattern method from testing robot's slope in degrees value while robot walks on even and uneven terrain. This test performed in ten times of iteration, with each of iteration using same forward command, and this test performed in 60cm route length of even and uneven terrain.

B. Testing Analysis

Figure 6 shows result of stability test. From this test obtained average slope on even terrain and uneven terrain is 4^{o}~6^{o} and 8^{o}~9^{o}, with this result is still below 10^{o} slope, this robot can walk fairly stable with no serious problems occur on performed test.

The stability of the robot in this test is influenced by many factors, including servo performance, the mechanical design of the robot, slippage between floor and robot's legs, and the values of variables at the input of the sine pattern.

V. DISCUSSION AND FUTURE WORK

Inverse kinematics combined with sine pattern can be effective for locomotion method on quadruped robot, because sine pattern would be a good solution to inverse kinematics problem to get groups of end effector position that we need on inverse kinematics calculation, however we still have to define some variable for each movement to obtain the desired locomotion model we need.

This robot locomotion system is still in an early stage, there are still missing important terrain adaptation systems such as sensors that can be used to stabilize slope on robot while it walks on uneven terrain. In the future, we would like the algorithm is combined with central gravity control and using sensor to enhance the locomotion stability and efficiency of the robot's movement.

REFERENCES

[1] Morimoto, Jun Endo, Gen Nakanishi, Jun Chen, gordon.2008. "Modulation of Sinusoidal Patterns by a Coupled Oscillator Model". Japan Sci. & Techno. Agency.

[2] Yamazaki, Kenneth S.1999. "The Design and Control of Scout I A Simple Quadruped Robot". Montreal :Department of Mechanical Engineering McGill University.

[3] Gonzalez de Santos, Pablo Garcia, Elena, and Estremera, Joaquin. 2006. "Quadrupedal Locomotion An Introduction to the Control of Four-legged Robots". London : Springer-Verlag London Limited.

[4] G.Morse, S.Risi, C.R.Snyder, K.O.Stanley.2013. "Single-Unit Pattern Generators for Quadruped Locomotion". New York : University of Central Florida.

[5] Wang, Lianqing Yu Yujin dan Tao, Weijun.2010. "Gait Analysis and Implementation of a Simple Quadruped Robot".China : Wuhan University of Science and Engineering and Nanjing University of Science and Technology.

[6] Endang.2013."WR note Electrical Engineering". [Online]. Available: https://wangready.wordpress.com. [Accessed 2016].

[7] Buehler, Martin.1998. "SCOUT: a simple quadruped that walks, climbs,and runs". Belgium : Walt Disney Imagineering.

[8] Tarokh, Mahmoud and Lee, Malrey. 2009. "Systematic Method for Kinematics Modeling of Legged Robots on Uneven Terrain". San Diego :Department of Computer Science, San Diego State University.

[9] Gómez, Javier Muñoz-Martínez, Pablo, R-Moreno, María D Romeral Julio J. López, Sara Navarro.2009."An Autonomous system for the locomotion of a Hexapod Exploration Robot".Madrid : Departamento de Automática, Universidad de Alcalá.

[10] Meng, Ni Xiaodong, Wang. 2011."Kinematics Analysis and Simulation of Quadruped Robot". Beijing: School of Automation Science and Electrical Engineering, Beihang University.