Implementation of Inverse Kinematics for the Coordination Control of Six Legged Robot

Wulandari[1], Karlisa Priandana[2], Agus Buono[3]

[1] Department of Computer Sciences, Faculty of Mathematics and Natural Sciences, Bogor Agricultural University, Bogor, Indonesia (wulandari.ilkom@apps.ipb.ac.id)

Abstract

Six legged robot, or hexapod, is a kind of legged robots with static stability. The robot has many variants of leg movements. This should be taken into consideration by the robot designer. The research is to adopt a leg movement algorithm named inverse kinematics. Inverse kinematics is used to find each leg angle required to produce certain position of the end-effector. The experiments show that pulse width input can move the robot faster than that with end-effector displacement distance input. However, the distance input is more flexible due to its simplicity such that the user can input the desired end-effector displacement distance without relying on the pre-determined distances. The inverse kinematics algorithm has been developed, although it is not able to avoid or circumvent the barriers.

Keywords: *inverse kinematics, hexapod, robot*

I. INTRODUCTION

Robot is tools that can be programmed or re-programmed and has a mechanical manipulator or actuator in the form of components or tools specific to the various programs that are easily adjustable to carry out various tasks. A tool can be categorized as a robot if the device has a minimum of 4 degrees of freedom (DOF). Robot mechanism consists of a rigid body system connected by joints. The position and orientation of the robot in space is called the pose. Robot kinematics described by pose, velocity, acceleration, and derived from the robot body pose motion generating mechanism (Siciliano 2008).

Based on the stability, the stability of legged robots can be divided into two categories, dynamic and static stability. Legged robot which includes the dynamic stability is a robot with an emphasis that does not affect the stability of the robot when it moves. In general, this type of robot has one to four legs. Legged robot which includes the static stability is a robot that the center of gravity remains stable when moving. This type of robots have at least 4 legs (Thirion and Thiry 2002), like the hexapod and the octopod.

Hexapod robot has variants of movements. With that amount of legs, resulting in the complexity of leg movements, the robot needs a corresponding algorithms that can make hexapod move according to environmental conditions at the time (Thirion and Thiry 2002).

The form of hexapod legs is included to open chain kinematics. Open chain kinematics is when the end- effector of the leg / arm robot is not connected back to the base of the leg joints. Robot mechanism can be calculated by forward kinematics if joint variables and velocity values are known. The position and orientation of the end - effector based on base frame is given by evaluating the transformation matrix consisting of the rotation matrix and translation matrix. Inverse kinematics is a joint decision variable based on the position and orientation of the end - effector (Lewis et al. 2003). Coordinating robot motion can be used with inverse kinematics.

II. METHODS

• *Study literature*

This stage is used to study the datasheet of the components and algorithm that will be used

• *Preparation of equipment and materials*

The components needed to assemble a six-legged robot is 12 servo HS645MG , 6 HS225BB servo, Arduino Uno microcontroller, SPC servo controller , and frame the robot MSR01.

• *Development of the robot structure*

Development step of the robot structure consist of coupling tools and materials and data retrieval robot workspace.

• *Development of robot movement algorithm*

This phase includes the development of data analysis and artificial intelligence workspace. Data workspace have been obtained from the previous step will be analyzed in order to produce the expected position of the leg tip.

• *Implementation of the robot*

At this step, the algorithms that have been developed will be implemented on a six-legged robot.

• *Testing and repair system*

Testing step is used to check the functionality and specifications of a six-legged robot that has been fitted with artificial intelligence. If the desired function and specification has not been reached, there will be improvements.

• *Evaluation of the system*

At this step, flaws and successes that have been achieved will be evaluated.

III. RESULTS AND DISCUSSION

Preparation tools and materials

Materials used in this study include:

- Frame of six-legged robot
- 12 pieces servo HS645MG
- 6 pieces servo HS225BB
- 1 piece SPC Servo controller
- 1 piece Arduino Uno microcontroller
- 9V Battery
- 7.4V 2200mAh LiPo Battery + 5.1V Regulator

The software used was 1.0.3 and Arduino using C language.

Development of robot structure

Each leg of the six-legged robot consists of 3 pieces servo is 2 pieces Servo HS 645MG and 1 servo HS 225BB. Servo position at the leg chain can be seen in Figure 2. HS 645MG servo and servo HS 225BB have room for 180 degree movement by changing the pulse width. 1500 microseconds pulse width gives 90 degrees servo position, pulse width 600 microseconds provide servo position 0 degrees, and 2400 microseconds pulse width gives 180 degrees servo position. Servo HS 225BB only used for swinging the leg forward and back, so it does not require a large operating power. In order to lifting weight, the robot required considerable strength to support the weight of the robot using servo HS 645MG. Robot coupling results can be seen in Figure 3.

Development of robot movement algorithm

Inverse kinematics is used to obtain the value of angle at each joint (Θ1, Θ2, and Θ3) based on the position information x, y, and z are given. Inverse kinematics is obtained by using the method of geometry.

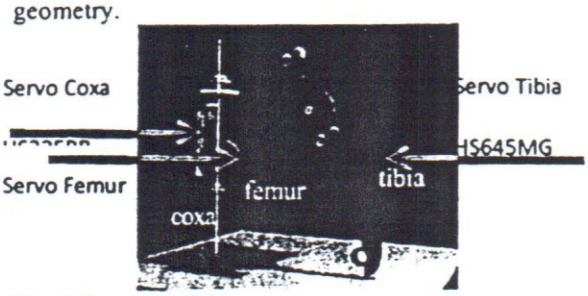

Fig. 1 Servo position

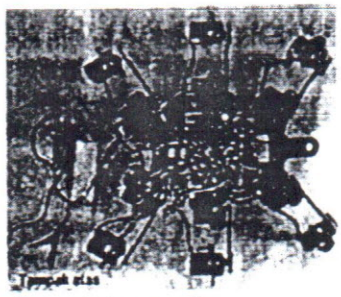

Fig. 2 Implementation result

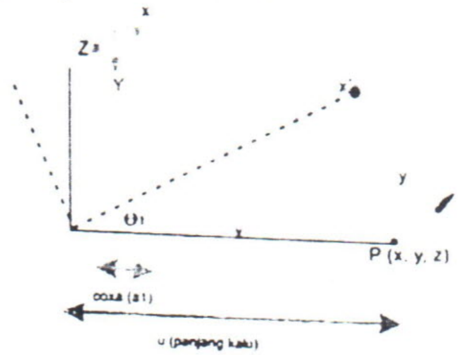

Fig. 3. Robot Geometry from above

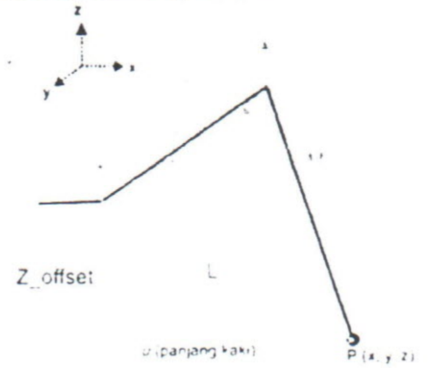

Fig. 4. Robot Geometry from front

Based on Figure 4, the Θ1 angle can be calculated as follows:

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

To calculate the angle Θ2 and Θ3 required the help of corner angles A1, A2 and A3 as shown in Figure 5. Angle A1, A2, and A3 can be calculated as follows:

$$A_1 = \tan^{-1}\frac{(L)}{Z\_offset} \quad (2)$$

$$A_2 = \cos^{-1}\frac{(a_3^2 - L^2 - a_2^2)}{-2\, a_2 L} \quad (3)$$

$$A_3 = \cos^{-1}\frac{L^2 - a_3^2 - a_2^2}{-2.\, a_3 . a_2} \quad (4)$$

Z_offset is the distance from the floor to the lower limit of the robot body. At the time of z = 0, Z_offset is 78 mm. For the other z positions, Z_offset can be expressed by the following equation:

Z_offset = 78 - z (5)

L is the length of the robot leg with the equation:

$$L = \sqrt{(u - \text{panjang coxa})^2 - y^2} \quad (5)$$

with u is the distance from base to coordinate end-effector and y is the position of the end-effector on the y axis.

Angle Θ2 and Θ3 in Figure 5 can be expressed as follows:

$$\theta_2 = 90 - (A_1 + A_2) \quad (6)$$

$$\theta_3 = 90 - A_3 \quad (7)$$

Angle calculation is different from the servo angle. On the servo, angle starts from 0 to 180 degrees, while the angle geometry calculations starting from -90 to 90 degrees. To normalize the leg, the teta2_degree_1 is subtracted by 54.85. This value is obtained from the calculation of the inverse kinematics angle position when the leg is in a stable position at 90 degrees. Normalization was also performed on teta3_degree_1 by a subtraction of 41.33, so the value of joint angle on each leg is at 90 degrees.

Implementation of the algorithm on Robot

Using inverse kinematics algorithm that has been developed is implemented on a six -legged robot that has the dimension limits as can be seen in Table 1. These limits are used to see the robot workspace.

Table 7 Workspace of robot

Joint	link	length (mm)	Min angle	Max angle
1	Coxa	15	-90°	90°
2	Femur	80	-90°	90°
3	Tibia	130	0°	90°

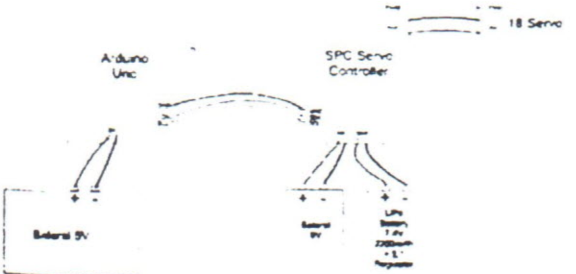

Figure 5 Robot Connections

Implementation of the system was performed as in Figure 5. Microcontroller will receive the input of motion commands (left, right, forward, and backward). Microcontroller connected to the servo controller using I2C connection. SDA port (Pin A4) and SCL port (Pin A5) on the microcontroller will be connected with SDA port and SCL port on SPC servo controller.

Table 8 Comparison of computational time of algorithm

Movement type	Distance (mm)	Rotary angle (°)	Time needed each step: Using band width	Time needed each step: *Inverse kinematics*
Forward	130	0	2.05	4.29
Backward	50	0	2.23	4.30
Left	0	45	3.42	5.44
Right	0	45	3.28	5.42

Movement phase from a six-legged robot consists of four poses are cyclical which are power, lift, swing, and contact. Power pose is when the leg position just above the surface and stable. Lift pose is when part of the femur and tibia joint rotates counter-clockwise so that leg lifts along the z axis. Swing pose is when coxa servo rotates along the spin axis z. Contact pose is when the leg returns from above until z axis is 0. These phases will continue to repeat so that the robot will move.

To move forward, 3 hexapod legs will move forward while the other legs stand still. After the movement of 3 legs has completed, the other 3 legs move the same way. This movement pattern is called a tripod gait.

Implementation of the algorithm is done in two ways:

1. With the input of the PWM pulse width value, using the result of the previous calculation of inverse kinematics.
2. With the input of the distance moved by the desired end-effector and included in the calculation of inverse kinematics algorithm.

Evaluation of the system

Table 2 shows that the computing time of the first algorithm is faster than the second algorithm. This occurs because of time calculation of inverse kinematics on the second algorithm. However, the second algorithm is more flexible because users can define their own end-effector displacement distance desired without relying on a pre-determined distance.

Overall, it can be seen that the developed inverse kinematics algorithm has been successfully implemented on a hexapod. This algorithm has fast enough computing time, especially when implemented by directly entering the pulse width value of inverse kinematics calculations. However, this algorithm is not able to adjust with the barrier. In other words, the movement of the robot legs will remain the same even though there is a hitch. FFSA algorithm developed by Reyneri (1997) can be implemented to overcome these shortcomings. However, since the calculation is more complex, the implementation of this algorithm would require a longer computation time.

IV. CONCLUSION

Calculation algorithm using inverse kinematics have been developed and implemented on a hexapod. Implementation of the algorithm with the input of the pulse width values have been calculated previously has faster time computation than the implementation of the algorithm with the input of end-effector displacement distance. However, the second algorithm is more flexible since the users can define their own end-effector displacement distance desired without relying on a pre-determined distance.

REFERENCES

[1] ATMega. 2009. 8-bit AVR Microcontroller with 4/8/16/32K bytes in-system programmable flash ATmega48PA ATmega88PA ATmega168PA ATmega328P [Internet]. [downloaded 2013 Mei 25]. Available at: http://www.atmel.com/Images/doc8161.pdf.

[2] Berardi F, Chiaberge M, Miranda E, Reyneri LM. 1997. A walking hexapod controlled by a neuro-fuzzy system. In Minnesota College Personnel Association. 1997 Feb 10-12; Pisa, Italia. Pisa (IT). hlm: 95-104. doi: 10.1.1.66.7520.

[3] Igoe T. 2010. Arduino Uno [Internet]. [downloaded 2012 Des 9]. Available at: http://www.arduino.cc/en/Main/arduinoBoardUno.

[4] Innovative Electronics. 2011. Manual SPC servo motor controller [Internet]. [downloaded 2012 Des 2]. Available at: http://www.innovativeelectronics.com/innovative_electronics/ pro_spcservo.htm.

[5] Lewis F, Dawson D, Abdallah C. 2003. Robot Manipulator Control Theory and Practice Ed ke-2. New York (US): Taylor and Francis.

[6] PTDesign. 2000. Pulse width modulation [Internet]. [downloaded 2013 Jun 23]. Available at: fab.cba.mit.edu/classes/MIT/961.04/topics/pwm.pdf.

[7] Reyneri LM. 1997. An introduction to fuzzy state automata. In International Work-Conference on Artificial and Natural Neural Networks. 1997 Jun 4-6; Canary Islands, Spanyol. Canary Islands (ES). pg: 273-283.

[8] RCToys. 2001. Announced specification of HS-645MG standard deluxe high torque servo [datasheet]. [downloaded 2012 Nov 2012]. Available at: www.rctoys.com/hitec-servos/HIT-HS645MG.pdf.

[9] RobotShop. 2001. Announced specification of HS225BB might mini servo [datasheet]. [downloaded 2012 Des 9]. Available at: www.robotshop.com/ PDF/HS225bb.pdf.

[10] Siciliano B, Khatib O 2008. Handbook of Robotics [bibliography]. Berlin (DE): Springer.

[11] Thirion B, Thiry L. 2002. Concurrent programming for the control of hexapod walking. ACM SIGAda Ada Letters. 22 (1):17-28. doi: 10.1145/507548.507551.