

Development of a Biped Walking Robot

Yong-Sung Kim^{*}, Chang-Jun Seo^{**}

^{*} Department of Electronics Engineering, Kyungpook National University, Daegu 702-701, Korea
(Tel : +82-53-940-8853 E-mail: gon2263@korea.com)

^{**} Department of Electronics Engineering, Inje University, Kimhae 621-749, Korea
(Tel : +82-55-320-3438 E-mail: elecscj@inje.ac.kr)

Abstract: In this paper, we introduce biped walking robot which can static walking with 22 degree-of-freedom. The developed biped walking robot is 480mm tall and 2500g, and 22 RC servo motors are used to actuate. Before made an active algorithm, we generated the motions of robot with the motion simulator which developed using by C language. The two dimension simulator is Based on the inverse kinematics and D-H transform. The simulator implements various motions as inputted the ankle's trajectory. Also we developed a simulator which is applied the principle of inverted pendulum to acquires the center of gravity. As we use this simulator, we can get the best appropriate angle of ankle and pelvis when the robot lifts up its one side leg during the working. We implement the walking motions which is based on the data(angle) getting from both of simulators. The robot can be controlled by text shaped command through RF signal of wireless modem which connected with laptop computer by serial cable.

Key words : biped walking robot, kinematics, principle of inverted pendulum

1. INTRODUCTION

As the personal computer in old times, the price of robot is gradually decreasing and many scientists said that the time will come soon that one family have one robot. The fact that the government selected the intelligent robot one of the 10 next generation of enterprise is speaking well the importance of robot industry. Among them, the research about humanoid robot has been processed actively. The humanoid robot has large value of practical use in the point of view that it has same task region with human. Japan, which is leading country in robot, advanced the technique more in humanoid robot like what ASIMO of HONDA and SDR of SONY. Currently, the development of the biped walking robot has been preceded actively, but truly do not reach the Japan yet.

We said that the biped walking robot is the robot which walks with 2 legs like human. In order to imitate the human's motion, generally each leg got 6 degree of freedom and controlled by microprocessor or microcontroller.

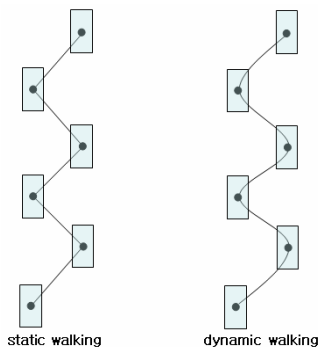


Figure 1. The path of center of gravity

The control algorithm is divided into static walking and dynamic walking. Each part of the path of Static walking has straight line as the motion pattern is acted sequentially, and the dynamic walking has curved path as the motion pattern is acted continuously. These control algorithms are various according to various forms of sensor interfaces and various theories.

In this paper, we propose to apply the principle of inverted pendulum to implement stable walking. The robot's motions are implemented with mechanical analysis except the notion of force, and implement more stable walking with the principle of inverted pendulum.

2. THE COMPOSITION OF ENTIRE SYSTEM AND OVERVIEW OF MOTION CONTROL

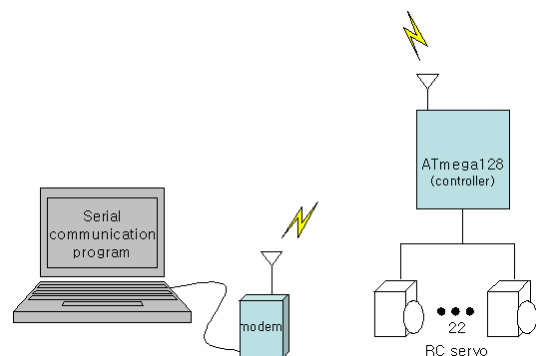


Figure 2. The entire system

The robot system is composed with the wireless modem which is connected with the laptop computer, robot's body and the controller which is loaded in the body of the robot. The

robot is controlled by text shaped command through RF signal of wireless modem which connected with laptop computer by serial cable. The robot conducts the pre-programmed action according to the received command.

The walking of the robot is implemented repeating the sequence like stretching right leg, moving center of gravity to the right and then stretching left leg, moving center of gravity to the left. The action of stepping forward is implemented by applying the angles that getting from 2 dimension motion simulator. And the result(angles) of simulation about one leg can apply to the opposite side leg. Because the sign of opposite side leg's angle is contrast. Thus we simulated about one side leg only. Also we developed a simulator which is applied the principle of inverted pendulum to acquire the center of gravity. The simulator about center of gravity acquires the best angle of ankle and pelvis when the robot moves the center of gravity to left and right side, and we applied the acquired angle to the robot control.

3. THE PROCESS OF DEVELOPMENT AND DESIGN

3.1 Setting up the coordinate system of each joint.

At first, we selected the degree of freedoms of each joint, and selected the suitable sequence of Yaw, Roll, and Pitch to combine organically. Then, we defined the center of each pelvis to a reference coordinate system, and the Z axis of each coordinate system of joints was set to the axis of rotation. The coordinate system of developed robot is set through the D-H representation and the content is as follows.

(DH 1) The axis X_1 is orthogonal to Z_0

(DH 2) Axis X_1 meets the axis Z_0

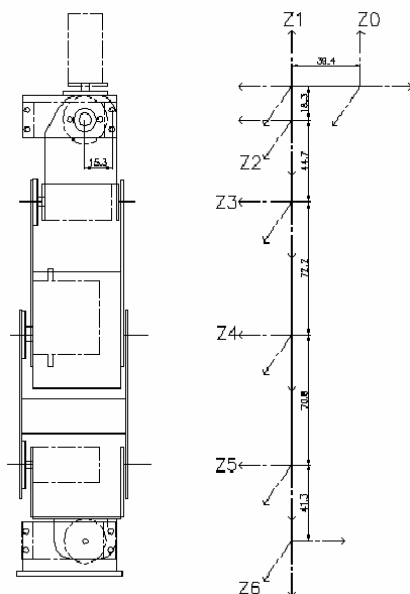


Figure 3. The coordinate systems of leg



Figure 4. Designed the body using Auto CAD

Table 1. D-H parameters of each joints

Joint	a_i	α_i	d_i	θ_i
2(Yaw)	0	-90	d2(-)	θ_2
3(Roll)	a_3	90	0	θ_3
4(Pitch)	a_4	0	0	θ_4
5(Pitch)	a_5	0	0	θ_5
6(Pitch)	a_6	0	0	θ_6
7(Roll)	a_7	-90	0	θ_7

The meaning of D-H parameters are as follows.

a_i : The distance between Z_{i-1} and Z_i (It exist when the two axis Z does not meet each other, It's Determined by the axis X_i)

α_i : The angle between Z_{i-1} and Z_i (It's Determined at the plane that is orthogonal to the axis X_i)

d_i : The distance between O_{i-1} , the origin of coordinate system $i-1$, and the Intersection of X_i and Z_{i-1} .

θ_i : The angle between axis X_{i-1} and axis X_i (It's Determined at the plane that is orthogonal to the axis Z_{i-1})

The reason why the parameter of the joint 1 does not exist in table 1 is that we set the R_0^1 to $\text{Tans}(d_1) \text{Rot}(z, -90)$.

When we set the center of each pelvis to the reference coordinate system, the transformation matrix of each link is as follows.

$$R_0^1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_1^2 = \begin{bmatrix} C_{\theta_2} & 0 & -S_{\theta_2} & 0 \\ S_{\theta_2} & 0 & C_{\theta_2} & 0 \\ 0 & -1 & 0 & -d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_2^3 = \begin{bmatrix} C_{\theta_3} & 0 & S_{\theta_3} & a_3 C_{\theta_3} \\ S_{\theta_3} & 0 & -C_{\theta_3} & a_3 S_{\theta_3} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_3^4 = \begin{bmatrix} C_{\theta_4} & -S_{\theta_4} & 0 & a_4 C_{\theta_4} \\ S_{\theta_4} & C_{\theta_4} & 0 & a_4 S_{\theta_4} \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_4^5 = \begin{bmatrix} C_{\theta_5} & -S_{\theta_5} & 0 & a_5 C_{\theta_5} \\ S_{\theta_5} & C_{\theta_5} & 0 & a_5 S_{\theta_5} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_5^6 = \begin{bmatrix} C_{\theta_6} & -S_{\theta_6} & 0 & a_6 C_{\theta_6} \\ S_{\theta_6} & C_{\theta_6} & 0 & a_6 S_{\theta_6} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_6^7 = \begin{bmatrix} C_{\theta_7} & 0 & -S_{\theta_6} & a_7 C_{\theta_7} \\ S_{\theta_7} & 0 & C_{\theta_7} & a_7 S_{\theta_7} \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The R_{i-1}^i means the matrix which express the i th joint's coordinate system regarding the $i-1$ th joint's coordinate system.

$$\begin{aligned} T_0^1 &= R_0^1 \\ T_0^2 &= R_0^1 R_1^2 \\ &\vdots \\ T_0^6 &= R_0^1 R_1^2 R_2^3 R_3^4 R_4^5 R_5^6 \\ T_0^7 &= R_0^1 R_1^2 R_2^3 R_3^4 R_4^5 R_5^6 R_6^7 \end{aligned} \tag{1}$$

By getting the $T_0^1 \sim T_0^7$, we can express and analyze each joint's coordinates with respect to the reference coordinate system.

3.2 Generating the motion with inverse kinematics.

The forward kinematics is a study to get the coordinate of end effector, on the other hand the inverse kinematics is to get the angle of joint which make the end effector coincide with the

arbitrary position we want. Inverse kinematics has various solutions unlike forward kinematics. But we can get the only one solution as we give the range to each axis of rotation.

There are various methods of inverse kinematics, but most of the methods are very difficult to use. So we use the iterative method in programming. Namely, the method is to find the most approximated solution with changing the each joint's angle one by one inside the range. The developed robot act static walking as repeating the following sequence, stretching right leg, moving center of gravity to the right, and then stretching left leg, moving center of gravity to the left. Thus, when the one side leg steps on, the stepping leg's angle of pelvis and ankle and the opposite side leg's angles of each joint do not change. That means we can reduce the variable when we think about the varying angle only.

Consequently, there remain only three variables to simulate, the θ_4 , θ_5 and θ_6 corresponding to pitch of stepping leg. If we assume that the sole is always parallel to surface, θ_6 can be expressed by combination of θ_4 and θ_5 . Namely, we can get the approximated solution as changing the θ_4 and θ_5 only. Hence, we can get (2) from figure 5.

$$\begin{aligned} 90 + \theta_6 &= \theta_5 + 90 - \theta_4 \\ \therefore \theta_6 &= \theta_5 - \theta_4 \end{aligned} \tag{2}$$

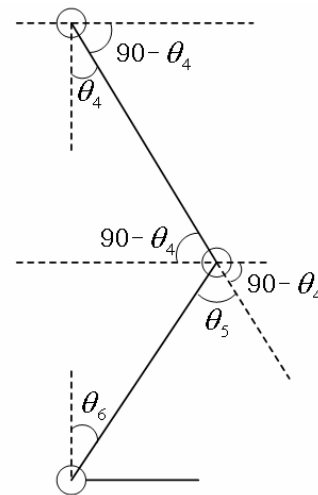


Figure 5. Explaining the elimination of θ_6

The last process is trajectory plan. The trajectory was set simply to second order function, and the formula of trajectory is expressed as follow.

$$y = -\frac{peak}{(distance/2)^2} \cdot x(x - distance) \tag{3}$$

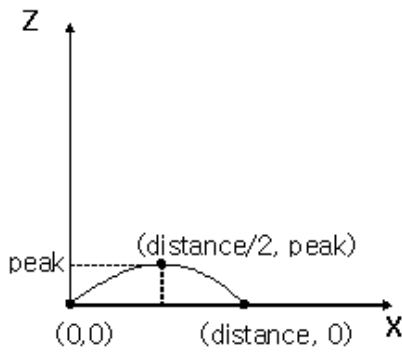


Figure 6. Setting of trajectory

We should execute repeatedly the (1) when trajectory was set to the tip of toes. But we can ignore the R_5^6, R_6^7 when we set trajectory about the ankle. Although the R_5^6 and R_6^7 are ignored, the result is same. Because, we assume that the sole is parallel to the surface.

We designed the simulator according to the settings and assumptions. Also we saved the result of coordinate of each joints, then display sequentially whether the motion acting smoothly.

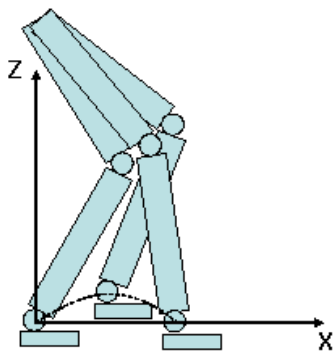


Figure 7. Trajectory plan

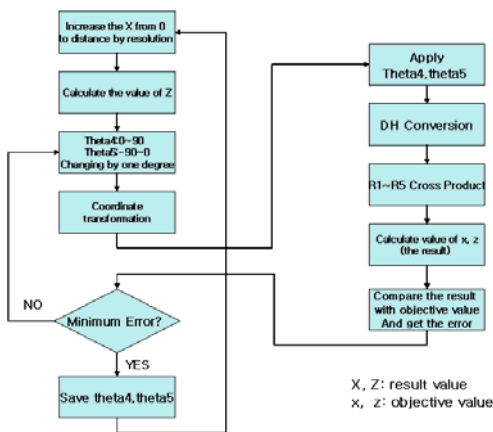


Figure 8. The algorithm of simulation

3.3 Applying and analyzing the principle of inverted pendulum for stabilization of robot while working.

The most important thing when implement the robot's walking is do not fall down, namely implement the stable walking. The moment which the robot comes to be most unstable is the moment which the robot lifts up its one side leg. In this paper, we try to apply the principle of inverted pendulum instead of try&error method for stable walking.

As simply each part of the body to one point mass and ignore the each of link's weight, we can assume the each masses to inverted pendulum which connected to the ankle with length l when the roll of ankle act to axis of rotation. And as we set simulation environment to 2 dimension coordinate whose origin is roll of ankle corresponding to axis of rotation, the center point of each masses are expressed to the coordinate which is about axis x and y. When each of point masses are defined, we can get the straight line l which connected to the origin and angle θ between l and axis y. If we assume the gravity acceleration g is regular, then the torque will be the function of m, l and θ . When we put the m to the constant, the torque can be calculated by only the point mass's coordinate.

As you see at figure 10, we can suppose that the position is the balanced position when the sum of torque of left and right side became equal. Namely, when disregarding inertia, we can assume that the projection point of center of gravity to axis x is in the origin.

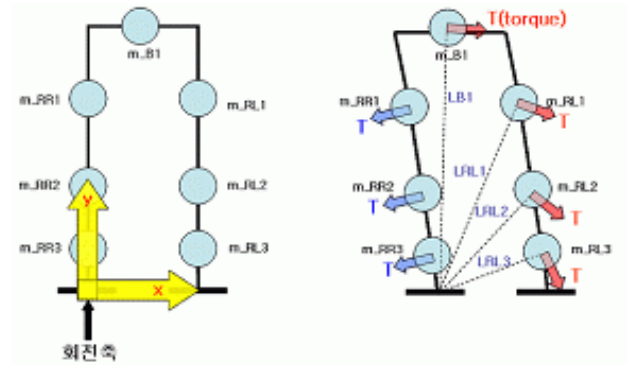


Figure 9. The torque of each point mass

In the walking process, the robot increase the angle of ankle and pelvis according to the roll to $|\theta|$ before it stretch its leg, at this time, we should find the optimized angle of $|\theta|$ to do not fall down Statically. When we assume the optimized angle is θ_s , the θ which holds (4) is the θ_s makes the robot to be the most stabilized.(Let θ_s to 'balanced angle')

$$\sum_{i=1}^n T_{R_i} = \sum_{i=1}^n T_{L_i} \tag{4}$$

According to the condition until now, we developed the simulator about center of gravity which can determine the

variance of left and right torques, and which can find the angle of roll of the ankle and pelvis to make the robot to be the most stabilized. The developed simulator about center of gravity does not use the transformation matrix for more rapid calculation. Transformation matrix is based on the 3 dimension. Because we do not tread the side of the robot, we set the coordinate of robot to 2 dimensions whose origin is roll of ankle corresponding to axis of rotation. And the process of calculating the coordinate of each point masses is explained at figure 11, 12 and 13.

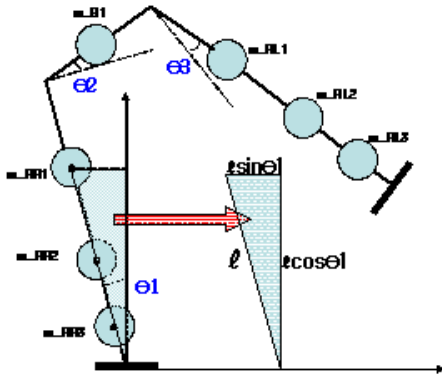


Figure 11. Deriving the coordinate of right leg

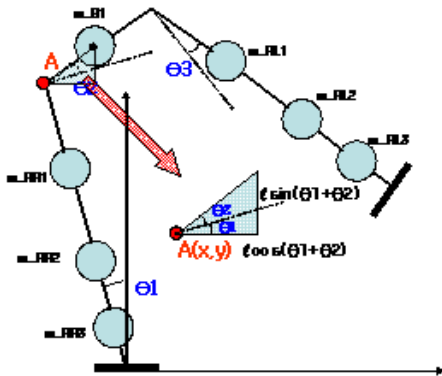


Figure 12. Deriving the coordinate of body

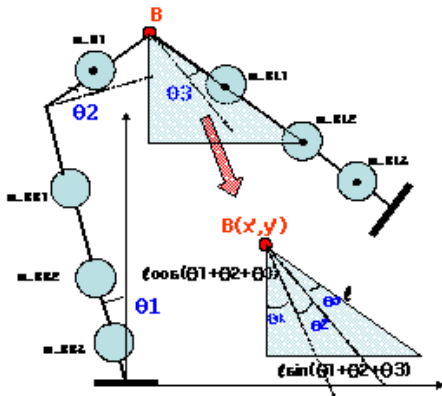


Figure 13. Deriving the coordinate of left leg

From the figure 11, 12 and 13, we can see that the coordinate of masses of right leg is determined by θ_1 , and the body

determined by θ_1 and θ_2 , and the left leg is determined by θ_1 , θ_2 and θ_3 .

Lastly, to find the torque of the point masses, there needs some process of linearization of the formula about torque T. The torque of inverted pendulum is as follow.

$$T = mgl \sin \theta \tag{5}$$

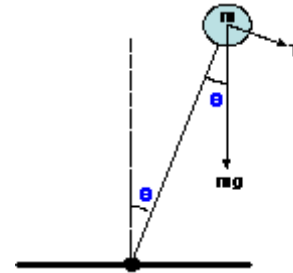


Figure 14. Torque of inverted pendulum.

The (5) can be linearization with using the term of first order of Taylor Series. If we set the m, g, l to constant and the torque is only function of θ , then, it can be expressed as follows.

$$\begin{aligned} T(\theta) &= T(\theta_0) + \frac{T'(\theta_0)}{1!}(\theta - \theta_0) \\ &= mgl \sin(\theta_0) + mgl \left[\frac{d \sin(\theta)}{d\theta} \right]_{\theta=\theta_0} \cdot (\theta - \theta_0) \end{aligned} \tag{6}$$

The (6) be the Maclaurin Series When we set θ_0 to zero, and then we can get the formula as follows.

$$\begin{aligned} T(\theta) &= mgl \sin(0) + mgl \cos(0)(\theta - 0) \\ &= mgl \theta \\ \therefore T &= mgl \theta \end{aligned} \tag{7}$$

According to (7), we calculated the torque of left and right side, and we can find the angle of roll and pelvis, namely θ_s , which makes the robot to be most stabilized.

3.4 22 servo motor control which use microcontroller AVR

The actuator of developed robot is digital servo and it can be controlled by periodic duty signal. There needs some process to input duty signal to many servos in limited number of timers, like what generating time-divided signal of the timer.

We use ATmega128 which is one of the microcontroller AVR, and we do not explain concretely about it. We will explain the algorithm of motor control only. The microcontroller ATmega128 has 2 8bit timers and 6 16bit timers. Among them, we use one 8bit timer and two 16bit timers to actuate both legs(12 motors) and body(2 motors). And the both arms(8 motors) are actuated by one 16bit timer. Eventually, 22 motors are actuated by 4 timers among 8 timers. The servo motor can be controlled by periodic duty signal which is time-divided.

4. EXPERIMENT AND SIMULATION



Figure 15. Forward short



Figure 16. Forward long



Figure 17. Backward

Table 2. The result of motion simulation

Time	Forward short		Forward long		backward	
	θ_4	θ_5	θ_4	θ_5	θ_4	θ_5
t	30	-60	0	-44	30	-60
t+ Δt	38	-72	12	-62	33	-72
t+2 Δt	44	-79	22	-75	34	-79
t+3 Δt	49	-83	31	-83	33	-83
t+4 Δt	53	-85	38	-87	31	-85
t+5 Δt	56	-85	44	-89	28	-85
t+6 Δt	57	-82	49	-87	24	-82
t+7 Δt	57	-78	52	-83	20	-77
t+8 Δt	54	-69	53	-75	15	-70
t+9 Δt	50	-58	50	-62	8	-59
t+10 Δt	43	-42	43	-42	0	-44

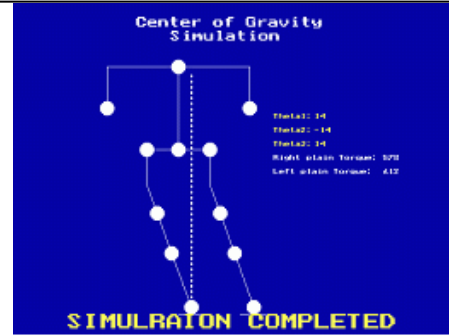


Figure 18. The result of center of gravity simulation

The table 2 is result of the motion simulation, and we can get $\theta_s=14$, the balanced angle, as the result of the simulation about center of gravity. But, there's some error when we apply the data of results to the robot. The biggest reasons are probably the mechanical freeplay and the backlash inside of motor. Especially, In the case of ankle and pelvis, the freeplay will be larger because of the excessive force, and then the walking is more unstable. We revised the mechanical freeplay with changing motor horn and strengthen the connection, also the backlash is revised by programming.

5. CONSLUSION

In this paper, we introduced biped walking robot which can static walking with 22 degree-of-freedoms. The developed robot can forward and backward walking and left and right turning, and it controlled by text shaped command which transmitted by wireless modem that connected by laptop PC. The developed simulator of motion and center of gravity can apply to the different size robot. If the degree of freedoms and the sequence of Yaw-Roll-Pitch are same, the different size robot can be applied to the simulators with changing some parameters(m and ℓ), and advances the time of implementation of the motion.

We should proceed the research about the control algorithm using various sensor like FSR(Force Sensing Resistor) and slope sensor for real-time revising the attitude, and also proceed the development of simulator which integrates the motion and center of gravity.

REFERENCES

- [1] Mark W.Spong, M. Vidyasagar, "Robot dynamics and control", John Wiley & Sons, 1989
- [2] Jang-Myeong Lee, jinyoung, "Introduction of Robotics", 2003
- [3] Young-Whui Sung, Su-young Lee, "The development of a miniature humanoid robot system", The institute of control automation and systems engineers, 2001.5
- [4] Young-Whui Sung, "A study on the inverse kinematics for a biped robot", The institute of control automation and systems engineers, 2003.12
- [5] Duck-Yong Yoon, "AVR ATmega128 master", Ohm, 2004