2018 18th International Conference on Control, Automation and Systems (ICCAS 2018)
Oct. 17–20, 2018; YongPyong Resort, PyeongChang, GangWon, Korea

# Development of a Transformable Self-Balancing Mobile Robot

Jinkwang Kim[1], Geonhoo Kim[2], Taewoo Han[2], Seungwon Heo[2], Yongjae Kim[3*] and Hyun Myung[4*]

[1]Department of the Robotics Program, KAIST,
Daejeon, 34141, Korea (jinkwang@kaist.ac.kr)
[2]Department of Electronics Engineering, Korea University of Technology and Education(Koreatech), Cheonan, 31253,
Korea (altn1374@koreatech.ac.kr, htw4551@koreatech.ac.kr, hsw8439@koreatech.ac.kr)
[3]Department of Electronics Engineering, Korea University of Technology and Education(Koreatech), Cheonan, 31253,
Korea (yongjae@koreatech.ac.kr)
[4]Department of Civil and Environmental Engineering, KAIST,
Daejeon, 34141, Korea (hmyung@kaist.ac.kr) * Corresponding author

**Abstract:** This paper reports the design, construction, and control of a mobile robot that can be transformed from the four-wheel mobile robot into two-wheel self-balancing robot and vice versa. The hardware of the robot utilizes the mechanism of the three-link manipulator. This robot is composed of three components; the body part, the middle link, and the top part. The system architecture comprises a pair of DC motor controllers to move the wheel, two servo motors to move the middle link and the top part, and an Arduino microcontroller board, etc. When the robot is transformed to the two-wheel self-balancing robot, the COM (Center of Mass) equation, and the IMU (Inertial Measurement Unit) sensor are employed for attitude determination. The method of the control is based on a proportional-integral-differential (PID) control. The results show the possibility of performing both functions of the four-wheel mobile robot and two-wheel self-balancing robot.

**Keywords:** Three-link manipulator, Two-wheel self-balancing robot, Center of mass.

## 1. INTRODUCTION

In recent years, the usage of the robot is expanding from industrial and military robot field to service robot field. The mobile robots have entered civilian and personal spaces such as schools, hospitals, and ordinary homes. Especially, wheeled mobile robots are widely used in virtue of their good mobility. The four-wheel mobile robot and the two-wheel self-balancing robot are two of the representative types of various mobile robots. The four-wheel mobile robots can be seen in daily needs in the form of delivery robot, cleaning robot, etc. In different circumstances, the two-wheel self-balancing robots are practically found as Segway personal transport, teleconference robot, etc.

The robot introduced in this paper is originated from the idea of combining these two types of robots. To transform the robot from the four-wheel mobile robot to the two-wheel self-balancing robot and vice versa, the robot applies the mechanism of the three-link manipulator. This robot is composed of three parts; the body part, the middle link, and the top part. The microcontroller and various electronic circuit parts for robot control are located in the body part. The two wheels operated by the DC motors are located at the end of this part and two passive omnidirectional wheels are located at the front side of it. The servo motor with belt reduction pulley drives the middle link. Another servo motor at the end of the middle link drives the top part. The top part has the battery and the tablet PC which are also used as a balance weight. The function of the robot is divided into the four-wheel mobile robot (driving mode), the two-wheel self-balancing robot (self-balancing mode).

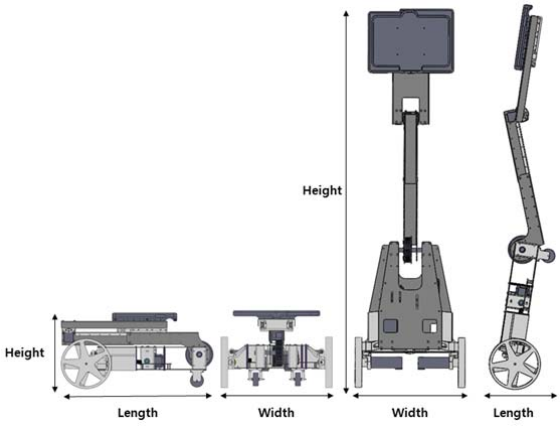


Fig. 1 CAD design of the robot.

To control the robot, The two-wheel self-balancing robot is investigated. It is very similar to the inverted pendulum, which is an important testbed in control education and research; see, for example, [1], [2]. Studies of the two-wheel self-balancing robots have been widely reported. For example, JOE [3] is early versions completed with inertial sensors, motor encoders and on vehicle microcontrollers. Since then, there has been active research on the control design for such platforms, including classical and linear multivariable control methods [3], [4], nonlinear backstepping controls [5], [6], and fuzzy-neural

control [7]. Alternatively, the Ziegler-Nichols method [8] is also used for control, when the dynamics model of the robot is unknown. Among the various approach, the heuristic method based on the Ziegler-Nichols method is adopted for finding PID gain values.

This paper is organized as follow: Section 2 describes the hardware and system architecture of the robot, Section 3 details the control, Section 4 presents the experimental results, followed by some conclusion and future works in section 5.

## 2. STRUCTURE OF THE ROBOT

Section 2 can be classified into three parts. Section 2.1 introduces the dimensions and the parts descriptions of the robot. Section 2.2 describes how to connect the electronic circuits. Section 2.3 discusses how to decide the specification of the servo motor and the DC motor.

### 2.1 Dimension and Parts descriptions

The dimension of the robot is divided into the driving mode and the self-balancing mode. In case of the driving mode, the dimension is the 330mm × 454mm × 229mm (Width × Length × Height). The dimension of the self-balancing mode is the 330mm × 225mm × 1130mm. The design of the robot in CAD can be seen in Fig. 1. Each mass of the body part, the middle link, and the top part is respectively 2974.81g, 523.28g, and 1629.07g. The diameter of the wheel to which the DC motor is connected is 150mm.

This robot uses the Arduino due as the main controller and the ATmega128 as the DC motor RPM (revolutions per minute) controller. To move the middle link and the top part, the servo motor is used. The sensors are the IMU and the encoder. In the case of the encoder, the two-multiplication circuit is constructed to increase the resolution of the encoder. The Bluetooth module is used to move the robot. The block diagram of the used parts can be seen in Fig. 2.

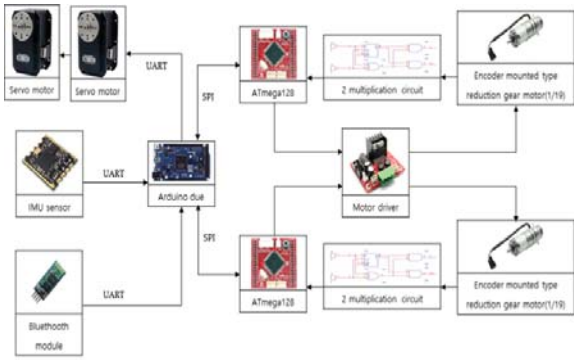



Fig. 2 Block diagram of the robot.

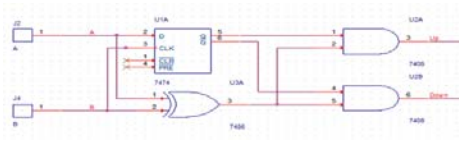
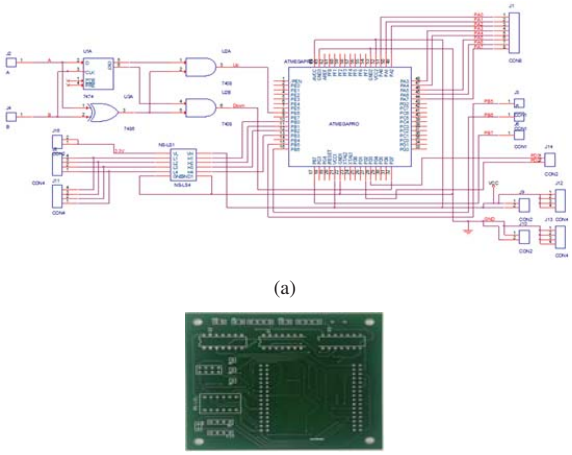


Fig. 3 OrCAD design of the two-multiplication circuit.



(a)

(b)

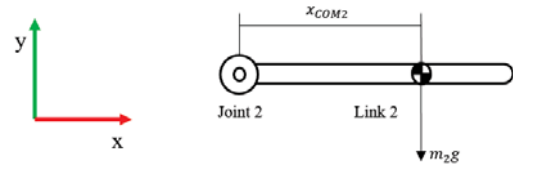
Fig. 4 (a) OrCAD design of the circuit, (b) PCB.

### 2.2 Electronic circuits

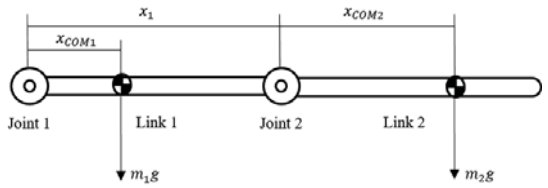
In the case of the Arduino due, The USART (Universal Synchronous Asynchronous Receiver Transmitter) communication is used to connect the servo motors, the IMU sensor and Bluetooth module to the Arduino due. The Arduino due sends the desired degree value to the servo motor and receives the raw degree value from the IMU sensor. It also receives data for the robot control from the Bluetooth module. To connect the ATmega128 of each wheel and one Arduino due, SPI (Serial Peripheral Interface) communication is used. The Arduino due is the master device and sends the desired RPM value to each ATmega128 as the slave device. It receives the current RPM value and the encoder value from each ATmega128.

The ATmega128 receives the data of the phase A and the phase B of the motor encoder using the Timer/Counter Interrupt. To increase the resolution of the encoder, The two-multiplication circuit is used. The two-multiplication circuit is made using the D flip-flop, the AND gate, and the XOR gate IC chip. Its design in the OrCAD can be seen in Fig.3.

The Timer/Counter interrupt of the ATmega128 is used to generate PWM (Pulse Width Modulation) and is connected to the motor drive. The motor drive then sends the PWM to the DC motor. To modularize the two-multiplication circuit and the ATmega128 used in the DC motor, the final electronic circuit was designed using the OrCAD and then the PCB is manufactured. The OrCAD design of the circuit and the PCB can be seen in Fig. 4.




(a)




(b)

Fig. 5 (a) Simplified robot to find the maximum required torque of the servo motor located at the end of the middle link, (b) Simplified robot to find the maximum required the torque of the servo motor in the body part.

### 2.3 How to Calculate Specification of Servo Motor and DC Motor

This section discusses how to decide the specification of the servo motor and the DC motor. When there is no movement of the robot, the specification of each motor is calculated in the bad situation.

In the case when the servo motor is located at the end of the middle link, as the top part extends horizontally, the maximum torque is required. Similarly, for the servo motor located at the body part, the maximum torque is required when the top part and middle link are horizontally extended. An example of these situations can be seen in Fig. 5.

In Fig. 5(a), The maximum torque's magnitude which is required to Joint 2 is as follows where $m_2$, $g$, $x_{COM}$, respectively, the mass of Link 2 and Joint 2, Gravitational acceleration, the COM of Link 2 and Joint 2.

$$\tau_2 = m_2 g x_{COM2}, \tag{1}$$

where $m_2$, $g$, $x_{COM2}$, respectively, the mass of Link 2 and Joint 2, Gravitational acceleration, the COM of Link 2 and Joint 2. Likewise, The maximum torque's magnitude required for Joint 1 in Fig. 5(b) can be obtained.

$$\tau_1 = (m_1 + m_2) g r, \tag{2}$$

where $r$, respectively, the COM of Link 1, Joint 1, Link 2, and Joint 2. $r$ is as follows

$$r = \frac{m_1 x_{COM1} + m_2(x_1 + x_{COM2})}{m_1 + m_2}. \tag{3}$$

Eq. (3) is substituted into Eq. (2) and summarized as follows.

$$\tau_1 = (m_1 x_{COM1} + m_2 x_1 + m_2 x_{COM2}) g. \tag{4}$$

The maximum required torque of the servo motor is obtained from Eq. (1) and Eq. (4), and the servo motor is decided by multiplying by the safety factor of 3.

To calculate the torque and the RPM of the DC motor in the transforming step of the robot or the self-balancing mode, The body part, the middle link, and the top part of the robot are regarded as one point mass. The torque and the RPM are calculated by choosing the situation when it is tilted 45° from the vertical axis of the ground as the bad situation. The simplified robot can be seen in Fig. 6.

When the robot is tilted by $\theta$ along the y-axis of the ground, it is assumed that the required acceleration to stand vertically is $a$. If the robot moves in the positive direction of the x-axis with the acceleration of $a$, it receives the force of $ma$ in the negative of the direction the x-axis by the inertia. At this time, the magnitude of $ma$ needs to satisfy the following inequality so that it can stand vertically.

$$\begin{aligned} mg\tan\theta \leqslant ma &= \frac{\tau}{r}, \\ mgr\tan\theta \leqslant \tau, & \end{aligned} \tag{5}$$

where $m$, $\tau$, $r$, respectively, the robot's mass, the required torque, and the wheel radius. Assign the above condition to $\theta$. Since the robot uses two DC motors, the inequality is reorganized for the $\tau_m$ of the motor torque.

$$\frac{mgr}{2} = \tau_m. \tag{6}$$

The required torque of the DC motor is obtained from Eq. (6). The reason that the final result is an identity is that the initial condition is the value considering the safety factor. If the robot is good at self-balancing, it will not tilt until 45° from the vertical axis.

According to the law of conservation of the kinetic energy, the potential energy of the robot is conserved as the kinetic energy. Using this law, the RPM can be obtained as follows,

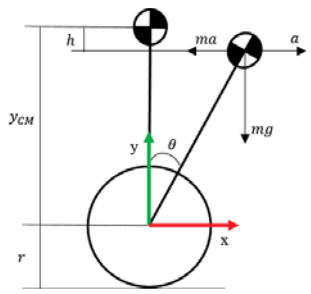



Fig. 6 Simplified robot to find the torque and the RPM of the DC motor

$$
\begin{aligned}
E_p &= mg(y - y\cos\theta), \\
E_k &= \frac{1}{2}mv^2 \quad , \\
v &= \sqrt{2gy(1-\cos\theta)},
\end{aligned} \tag{7}
$$

where $y$ and $v$, respectively, the robot's height when parallel to the y-axis, and the linear velocity of the wheel. The initial condition 45° is substituted into the result of Eq. (7). Then, the angular velocity is obtained from the relationship between the linear velocity and the angular velocity and converted into the RPM as follow,

$$
w = \frac{\sqrt{2gy(1-1/\sqrt{2})}}{r}, \tag{8}
$$

$$
RPM = w \times \frac{1}{2\pi} \times 60. \tag{9}
$$

Then, the required RPM of the DC motor is obtained from Eq. (9).

## 3. CONTROL

Section 3 can be classified into two parts. Section 3.1 introduces the calculation of robot's COM. Section 3.2 describes how the feedback loop is designed.

### 3.1 Center of Mass Calculation

This session describes how to calculate the COM of the entire robot when the robot is transformed. In Fig. 7, the coordinates of the body part, the middle link, and the top part are calculated from the reference coordinates using a homogeneous transformation matrix. The reference coordinate is the center of the wheel.

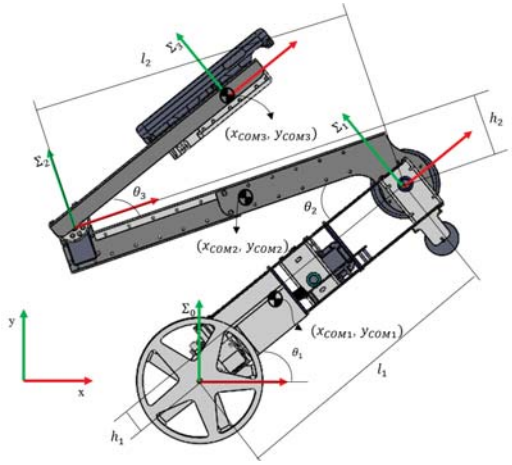



Fig. 7 Coordinate of each part. Where 1, 2, and 3, respectively, the body part, the middle link, and the top part.

The homogeneous transformation matrix $T_0^1$, $T_1^2$, and $T_2^3$ are obtained as follows,

$$
\begin{aligned}
T_1^0 &= \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & l_1 \\ -\sin\theta_1 & \cos\theta_1 & 0 & h_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
T_2^1 &= \begin{bmatrix} \cos\theta_2 & \sin\theta_2 & 0 & -l_2 \\ -\sin\theta_2 & \cos\theta_2 & 0 & h_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
T_3^2 &= \begin{bmatrix} \cos\theta_3 & \sin\theta_3 & 0 & x_{CM3} \\ -\sin\theta_3 & \cos\theta_3 & 0 & y_{CM3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{10}
$$

Using Eq. (10), the COM's coordinate of 1, 2, and 3 from the reference coordinates are obtained as follows,

$$
\begin{aligned}
P_{CM1}^0 &= T_1^0 \begin{bmatrix} x_{CM1} \\ y_{CM1} \\ 0 \\ 1 \end{bmatrix}, \\
P_{CM2}^0 &= T_1^0 T_2^1 \begin{bmatrix} x_{CM2} \\ y_{CM2} \\ 0 \\ 1 \end{bmatrix}, \\
P_{CM3}^0 &= T_1^0 T_2^1 T_3^2 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.
\end{aligned} \tag{11}
$$

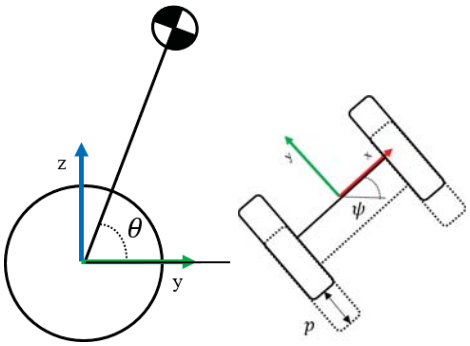
In each matrix of Eq. (11), the first row is the x-coordinate and the second row is the y-coordinate. I substitute these as $x_{CM1}^0$, $x_{CM2}^0$, $x_{CM3}^0$, $y_{CM1}^0$, $y_{CM2}^0$, and $y_{CM3}^0$. Then, the COM of the whole robot is obtained as follows,

$$
\begin{aligned}
x_{CM}^0 &= \frac{m_1 x_{CM1}^0 + m_2 x_{CM2}^0 + m_3 x_{CM3}^0}{m_1 + m_2 + m_3}, \\
y_{CM}^0 &= \frac{m_1 x_{CM1}^0 + m_2 x_{CM2}^0 + m_3 x_{CM3}^0}{m_1 + m_2 + m_3}.
\end{aligned} \tag{12}
$$

Where, $m$, $x_{CM}^0$, and $y_{CM}^0$, respectively, the mass of each part, the x-coordinate of whole robot, and the y-coordinate of whole robot. Then, the required length values are obtained by using the properties of the SolidWorks and applied to the calculated equations. And when the robot is operated, each angle is obtained by the IMU sensor and the servo motor. The COM of whole robot is used to control the robot.

### 3.2 Feedback Loop

The PCB in Fig. 4 connected to each wheel is the P controller that controls the RPM of the motor, and the control period is 500us. The PD control can show fast response speed, however, it is not used because it causes the vibration and the overshoot by the D. In the case of




(a) (b)

Fig. 8 (a) Simplified the right side view of the robot, (b) Simplified the above side view of the robot. Where $\theta$, $\psi$, $p$, respectively, the angle between the y-axis of the reference coordinates and the robot (Roll), the rotation angle of the robot (Yaw), and the position of the robot.

the Arduino due as the main controller, the control period is 5ms and the three values in Fig. 8 are used for the control. The roll is obtained by (12) and is as follows,

$$\theta \quad = \quad atan2(z^0_{CM}, y^0_{CM}). \tag{13}$$

The reason for $z^0_{CM}$ and $y^0_{CM}$ is that the coordinates system shown in Fig. 7 and Fig. 8 is expressed differently. The odometry from the encoder sensor is obtained and the values of the robot's position and the yaw are calculated using the odometry. The robot's roll and position are controlled by the PD controller since they require the fast response for the self-balancing. In the case of the yaw, the P control without the vibration is used even though the response speed is slow since it determines the heading of the robot rather than the self-balancing. For the right wheel, add the RPM values obtained through the roll PD controller, the robot's position PD controller, and the yaw P controller, then set this as the desired RPM of the right wheel. However, in the case of the left wheel, add the other RPM values and subtract the RPM obtained through yaw P controller. This is because RPM of the yaw P controller is related to the robot rotation. Then, set this as the desired RPM of the left wheel. The final controller is shown in Fig. 9.

Table 1 P AND D GAIN VALUE ACCORDING TO THE ROBOT OF POSTURE

| Step of posture | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Angle of the middle link | 45 ° | 45 ° | 125 ° | 160 ° |
| Angle of the top part | 0 | -125 ° | -125 ° | -160 ° |
| COM height | 0.14 m | 0.29 m | 0.36 m | 0.38 m |
| Roll P gain | 20 | 20 | 20 | 20 |
| Roll D gain | 0.35 | 0.35 | 0.425 | 0.35 |
| Position P gain | 180 | 200 | 220 | 220 |
| Position D gain | 150 | 100 | 75 | 85 |
| Yaw P gain | 20 | 20 | 20 | 20 |

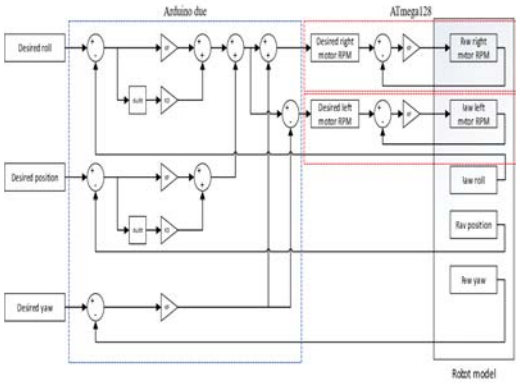



Fig. 9 Block diagram of controller

Fig. 10 Robot transformation process

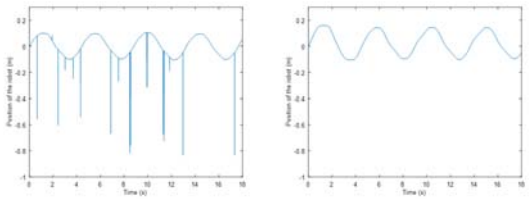
## 4. EXPERIMENTAL RESULTS

The process of the transforming from the driving mode to the self-balancing mode and otherwise is divided into five steps. In Fig. 10 from the left, it is the first step of posture. The gain values of the robot's posture from the second step to the fifth step are shown in Table 1. After, these values are adjusted linearly along the COM height of the robot, where, the COM height is $z^0_{CM}$ in Eq. (13).

In each step, We confirmed that it is self-balancing or not. However, there is the high frequency data in the position data. To solve this situation, the Low-Pass Filter (LPF) is applied to the robot's position data. For the performance experiments, the position data with LPF and the position data without LPF are compared in five step of Fig. 10. In Fig. 11, it can be confirmed that the high frequency data disappears after applying the LPF.

The way in which the robot is transformed from the driving mode to the balancing mode is as follows. Move the top part, until the IMU sensor that is mounted on the body part tilts more than 2° from the ground. After, The robot starts the self-balancing. The process of the transforming proceeds from the first step to five step in Fig. 10.

The algorithm transforming from the driving mode to the self-balancing mode is the opposite of the previous algorithm. In other words, the process of the transforming proceeds from five step to the first step in Fig. 10. However, there is the difference in the final landing process. In the final step, it is not clear which direction the robot will land in. It can be either forward or backward. To solve this, the RPM of the robot wheel is applied in the



(a) (b)

Fig. 11 (a) Robot's position without LPF. (b) Robot's position with LPF

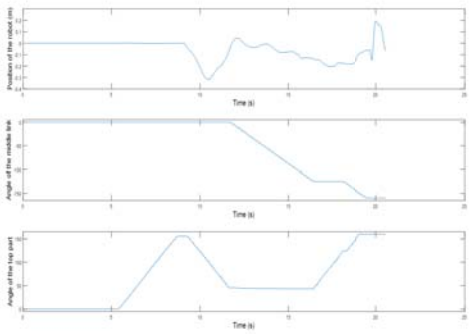


Fig. 12 Robot's position, middle link's angle, and top part's angle when the robot is transformed

minus direction. Then, The inertial force causes the robot to land in the plus direction and change into the driving mode.

When the robot is transformed from the driving mode to the self-balancing mode, the robot's position, the middle link's angle, and the top part's angle can be seen in Fig. 12. The angle of the middle link and the top part is input by the main controller. However the angle of the body part is received by the IMU sensor, just the passive value, therefore it is not considered in Fig. 12. The robot is transformed within the distance of $\pm$30cm based on the current position of the robot, and the transforming time is about 21s.

## 5. CONCLUSION AND FUTURE WORKS

In this paper, we propose a new type of mobile robot. In the driving mode, the robots height is only about 23 cm. Due to the four-wheel and low center of mass, the robot is possible to move with the stable posture. Thus it can be useful for moving on the uneven ground. On the other hand, in the self-balancing mode, the robot can stand up more than 1 m and interact with the human using the tablet PC on the top of the body. If this robot is improved a little more, the robot will be able to perform the delivery robot, the indoor guide robot, and the video teleconference robot.

In order to enhance the development of this robot, the following action could be done. The heuristic method is used without considering the robot dynamics model to determine the gain value. Therefore, in the future, we would like to design robot dynamics modeling to find gain values. Then, using the V-REP, The robot dynamics modeling is verified. The CAD design of the robot is in the V-REP. It can be seen in Fig. 13. In addition, current or torque sensors will be used to improve control accuracy.

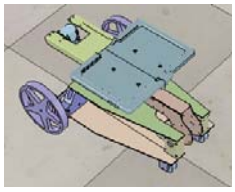


Fig. 13 Robot model in V-REP

## 6. ACKNOWLEDGEMENT

The student is supported by Korea Ministry of Land, Infrastructure and Transport (MOLIT) as U-City Master and Doctor Course Grant Program.

## REFERENCES

[1] R. Fierro, F.L. Lewis, and A. Lowe, "Hybrid control for a class of underactuated mechanical systems," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 29, No 6, pp. 649-654, 1999.

[2] Ke Xu and Xu-Dong Duan, "Comparative study of control methods of single-rotational inverted pendulum," in *Proc. of Int'l Conf. on Machine Learning and Cybernetics*, Vol. 2, Bejing, China, Nov. 4 - 5, 2002.

[3] F. Grasser, A. D'Arrigo, and S. Colombi, "JOE: a mobile, inverted pendulum," *IEEE Transactions on industrial electronics*, Vol. 49, No. 1, pp. 107-114, 2002.

[4] R. C. Ooi, "Balancing a two-wheeled autonomous robot." *University of Western Australia*, Vol. 3, 2003.

[5] T. Nomura, Y. Kitsuka, and T. Matsuo, "Adaptive backstepping control for a two-wheeled autonomous robot," *ICCAS-SICE*, pp. 4687-4692, 2009.

[6] N. G. M. Thao, D. H. Nghia, and N. H. Phuc, "A PID backstepping controller for two-wheeled self-balancing robot." *Int'l Forum on Strategic Technology*, Ulsan, South Korea, Oct. 13 - 15, 2010.

[7] K.-H. Su and Y.-Y. Chen, "Balance control for two-wheeled robot via neural-fuzzy technique," in *Proc. of SICE Annual Conference*, Taipei, Taiwan, Aug. 18 - 21, 2010.

[8] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *trans. ASME*, Vol. 64, No. 11, 1942.