2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

Rollover prevention by Quadruped Tracked Mobile Robot

Toyomi Fujita[1] and Shun Sato[2]

***Abstract*— A tracked mobile robot with legs is able to avoid rollover when moving on uneven terrain using its legs. A quadruped tracked mobile robot used in this study can quickly control the legs mounted on the four corners of the body so that they posture properly according to its tipping situation. In this paper, an inertial measurement device is mounted on the robot and posture information is obtained. Based on the obtained posture information, the robot can determine whether it is stable or not. We utilize the normalized energy stability margin to estimate the stability of the robot. The joint angles for the optimal leg posture to recover stability are computed by a simulation. Based on the result of the simulation, the appropriate rollover avoidance motion of the robot was obtained, and its effectiveness was confirmed in the fundamental experiments by the robot.**

I. INTRODUCTION

In recent years, the robots which can perform a work such as handling task have been expected. As a robot which is possible to work, we have developed a quadruped tracked mobile robot with two driving tracks and four active legs which can be used as working arms [1]. However, because the field where such robots works is generally consists of uneven and rough terrain, the situation of rollover may easily happen for the robot. In this study, we consider a method for rollover prevention by the robot. If the robot can obtain its posture information, it would be possible to prevent the rollover by controlling four legs properly. Some methods for rollover prevention have been presented for tracked robots by attaching sub-tracks and/or a manipulator [2][3]. The quadruped tracked mobile robot presented in this study can generates more stable postures using four legs so that it becomes optimal for rollover prevention according to situation.

This paper presents how to avoid the rollover for the robot using legs based on the information of IMU (Inertial Measurement Unit) sensor mounted on the robot. We also mention the methods for estimating stability of the robot to prevent rollover by detecting posture information, the center of rotation, the axis of rotation, and the rotation angle around it. Some fundamental experiments to confirm their validity are also reported.

II. QUADRUPED TRACKED MOBILE ROBOT

The authors have developed a quadruped tracked mobile robot [1]. Fig. 1 shows an overview of the robot. Two driving tracks are installed at the right and left side of the body. Four legs which have four degrees-of-freedom mechanism are also equipped at the corner of the body. The IMU sensor WT901, produced by WitMotion Co. Ltd., is mounted at the center of the body to obtain angular velocity and acceleration of the robot. Angular acceleration and tilt angle of the robot can be computed by difference and integration of the angular velocities around X-Y-Z axes respectively. Then state of the rotation of the robot is obtained from them.

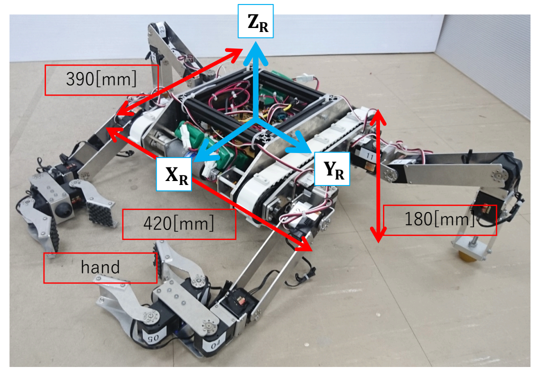

Fig. 1. Overview of developed quadruped tracked mobile robot

III. DETECTION OF POSTURE INFORMATION

A. Center of Rotation

Fig. 2 shows the status of the robot when it rotates the angles θ_x on the axis X_R and θ_y on the axis Y_R in the robot coordinate system Σ_R. Let us suppose that the position and axes of IMU sensor are the same as Σ_R. Using the measurement values of angular velocities of the rotation around X_R and Y_R, ω_x and ω_y, and translation acceleration on X_R, Y_R, and Z_R, a_x, a_y, and a_z, the center of ration $\boldsymbol{C} = (C_x, C_y, C_z)$ can be computed as follows.

With respect to the rotation around X_R on the Y_R-Z_R plane, the rotation angle θ_x can be obtained by

$$\theta_x = \int \omega_x dt. \tag{1}$$

Let a'_y and a'_z be Y and Z components of translation acceleration values generated by the rotation itself. These become

$$a'_y = a_y - g \sin\theta_x \tag{2}$$

and

$$a'_z = a_z - g \cos\theta_x \tag{3}$$

respectively, where g is gravity acceleration. Then, let θ_1 be the angle from Y_R to the vector (a'_y, a'_z) on the Y_R-Z_R plane,

*This work was not supported by any organization

[1]Toyomi Fujita is with Department of Electrical and Electronic Engineering, Tohoku Institute of Technology, Sendai 982-8577, Japan t-fujita@tohtech.ac.jp

[2]Shun Sato is SWS East Japan, Ltd., Japan

and r_x be the distance between the sensor and the center of rotation on the plane. They are given by

$$\theta_1 = \tan^{-1}\frac{a'_z}{a'_y}, \tag{4}$$

and

$$r_x = \frac{1}{|\omega_x|}\int\sqrt{{a'_y}^2 + {a'_z}^2}dt, \tag{5}$$

respectively. From these, we obtain the Y and Z values of the rotation center as

$$C_y = r_x \sin\theta_1 \tag{6}$$

and

$$C_z = -r_x \cos\theta_1. \tag{7}$$

With respect to the rotation around Y_R on the X_R-Z_R plane, we can also compute in the same way;

$$\theta_y = \int \omega_y dt, \tag{8}$$

$$a'_x = a_x + g\sin\theta_y, \tag{9}$$

$$\theta_2 = -\tan^{-1}\frac{a'_z}{a'_x}, \tag{10}$$

and

$$r_y = \frac{1}{|\omega_y|}\int\sqrt{{a'_x}^2 + {a'_z}^2}dt, \tag{11}$$

where θ_y is the rotation angle around Y_R, a'_x is the X component of the translation acceleration generated by the rotation, θ_2 is the angle from X_R to the vector (a'_x, a'_z) on the X_R-Z_R plane, and r_y is the distance between the sensor and the center of rotation, respectively. Then, the X value of the rotation center is given as

$$C_x = -r_y \sin\theta_2. \tag{12}$$

B. Axis and Angle of Rotation

We can obtain the rotation axis $\boldsymbol{n}$ and the rotation angle β around the axis using Rodrigues' formula by converting the rotational matrix given by θ_x and θ_y.

The rotation angle β is computed by

$$\beta = \mathrm{atan2}(\sqrt{a^2 + b^2 + c^2}, d) \tag{13}$$

where

$$c = \sin\theta_x \sin\theta_y,$$

$$b = \sin\theta_y(1 + \cos\theta_x),$$

$$a = \sin\theta_x(1 + \cos\theta_y),$$

and

$$d = \cos\theta_y + \cos\theta_x + \cos\theta_x\cos\theta_y - 1.$$

Then,the rotation axis $\boldsymbol{n}$ is given by

$$\boldsymbol{n} = \frac{1}{2\sin\beta}\begin{pmatrix} a \\ b \\ c \end{pmatrix}. \tag{14}$$

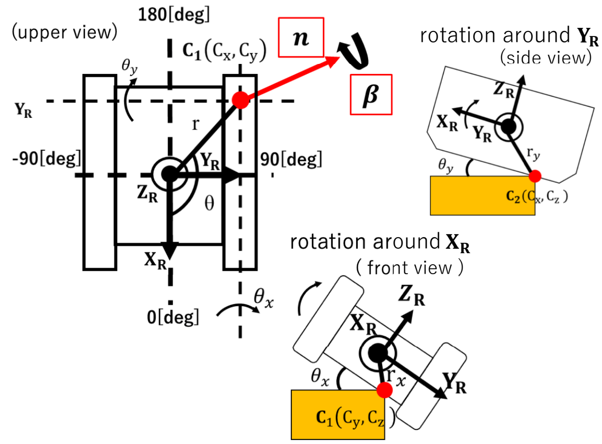

Fig. 2. Rotation of robot

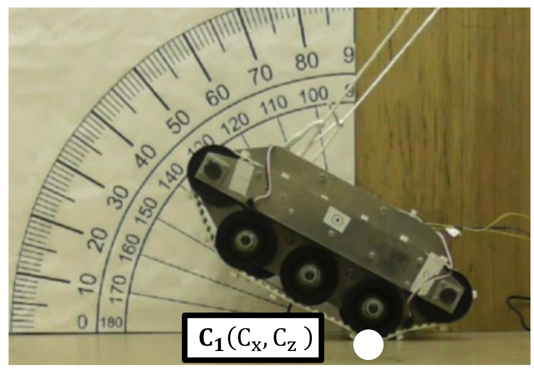

Fig. 3. Experimental setup for the rotation detection of robot

C. Experiments

We conducted fundamental experiments for the detection of the information of rotation. Fig. 3 shows an overview of the experimental setup. The robot with the IMU sensor was hanged by a string and rotated around $\mathrm{Y_R}$ axis only. The angle and angular velocity were obtained in the rotation. The rotational motion was simultaneously recorded as movie and the true rotation angles were extracted by the video frames. The rotational center on the X_R-Z_R plane was $\boldsymbol{C}_1 = (-80.4, -72)$ [mm].

Fig. 4 shows the result of rotational angle θ_y. Because the measured values are almost same as the true value, we can see the presented method for getting rotation angle is valid.

Fig. 5 shows the result of rotational center detection: X position of the center C_x to the rotation angle around Y axis θ_y. The obtained values have a large differences to true value of the rotational center $C_x = -80.4$ [mm]. We consider that it occurred due to unstable acceleration and angular velocity measured by the IMU sensor. Moreover, small errors were accumulated in computation in getting r_x and r_y.

Table I shows the detected rotation axis $\boldsymbol{n} = (n_x, n_y, n_z)$ and the angle of rotation β. The axis $\boldsymbol{n}$ is normalized as an unit vector. Thus, the n_y only should have value in this

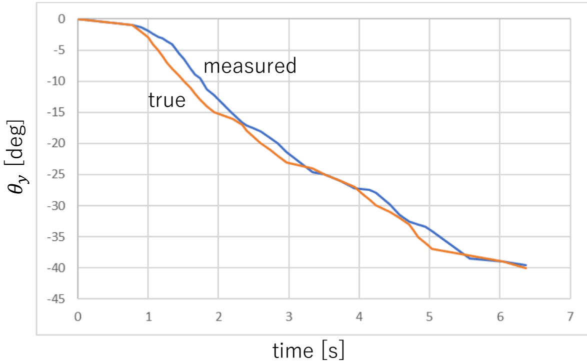

Fig. 4. Result of rotation angle detection around Y axis

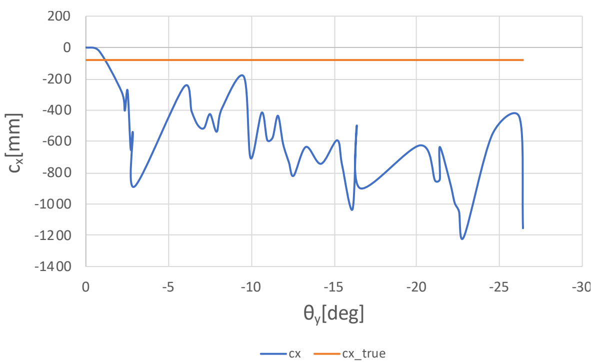

Fig. 5. Result of rotation center position on X to the rotation angle around Y axis

experiment and $\boldsymbol{n} = (0, 1, 0)$ because the robot is rotated around Y axis. The angle β should also be the same value to the rotation angle θ_y. The result shows valid values. However, it is the case of simple situation and there are large errors on the detection of the center of rotation as we mentioned above. So it may be difficult to determine correct center and axis in this method. We now consider simple rotations around Y_r or X_r only and assume the center of rotation is the edge point of the robot. We then focus on stability estimation and rollover prevention by using only the rotation angle sensed by the IMU sensor in the following sections.

IV. STABILITY ESTIMATION

We need to estimate stability in order to determine proper posture to avoid rollover for the robot.

The authors have considered two indices to do it: rollover stability margin [4] and normalized energy stability margin, called NE stability margin, [5]. The index values were

TABLE I

RESULT OF THE ROTATION AXIS AND ANGLE

true value θ_y[deg]	n_x	n_y	n_z	β[deg]
-10	-0.09	0.99	0.01	-10.0
-20	-0.05	0.99	0.01	-19.99
-30	0.00	0.99	0.00	-30.06
-40	0.00	0.99	0.00	-40.00

obtained in the experiment described in Section III-C and compared. As the result, there is no significant difference on the accuracy. Thus, we decided to use NE stability margin because we can compute it more simply, so it is suitable for performing by the robot.

Fig. 6 shows concept of NE stability margin, S_{NE}, by a rotation of the robot. S_{NE} is represented as

$$S_{NE} = h_{max} - h_0 \tag{15}$$

where h_0 is the height in the trajectory of the mass center of the robot, and h_{max} is the maximum of the height. It is more stable as S_{NE} is larger and the robot is almost the state of rollover when $S_{NE} = 0$.

V. ROLLOVER PREVENTION

A. NE Stability Margin Computation

In order to obtain the stability margin, S_{NE}, described by (15), we compute h_{max} and h_0. We now consider the rotation around X axis and assume that the center of rotation is the edge point of the robot.

Fig. 7 shows the front view of the rolling states in the rotation of robot. Initially, the robot rotates in the right direction around the edge point of the robot, $\boldsymbol{r}_1$, in the state of Fig. 7 (a). As the amount of rotation angle increases, the tip of inside right leg touches the ground at the state of (b). After that, the robot rotates around the tip position of the inside right leg in the state of (c). Therefore, the radius of rotation in this state becomes the maximum height of the mass center, h_{max}.

In the state of Fig. 7 (a), before the tip of leg touches the ground, the mass center position $\boldsymbol{g}_r(\theta) = (y_r(\theta), z_r(\theta))^{\mathrm{T}}$ to the rotation angle θ can be computed as

$$\boldsymbol{g}_r(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} y_r(0) - r_{1y} \\ z_r(0) \end{pmatrix} + \begin{pmatrix} r_{1y} \\ 0 \end{pmatrix} \tag{16}$$

where r_{1y} is Y value of $\boldsymbol{r}_1$. Therefore,

$$h_0(\theta) = z_r(\theta) \tag{17}$$

in this state.

When the tip of leg touches the ground at the rotation angle θ_α at the state (b), the mass center position becomes $\boldsymbol{g}_r(\theta_\alpha)$. Thus,

$$h_{max} = \sqrt{y_r(\theta_\alpha)^2 + z_r(\theta_\alpha)^2}. \tag{18}$$

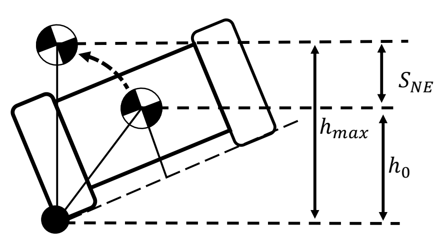

Fig. 6. Concept of S_{NE} by a robot rotation

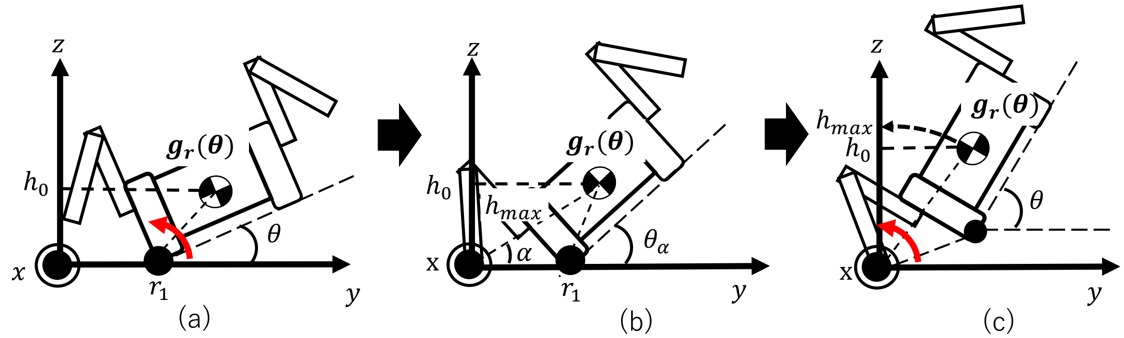

Fig. 7. States of the robot for rolling angles (front view)

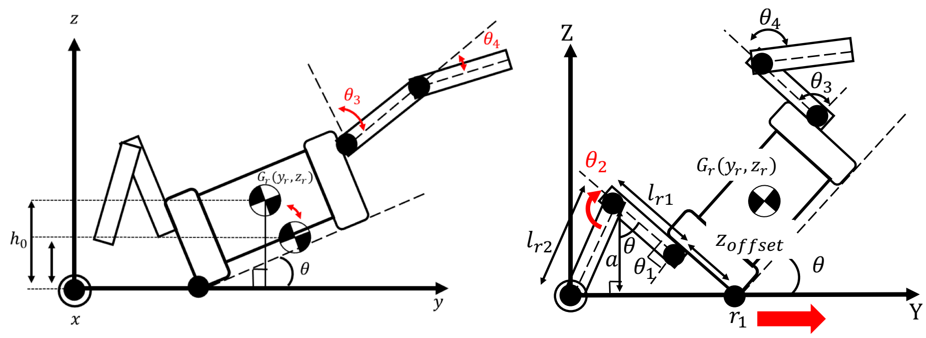

Fig. 8. Concept of the action for rollover prevention (front view)

In the state (b) and (c), because the position of rotation center changes,

$$h_0(\theta) = \sqrt{y_r(\theta_\alpha)^2 + z_r(\theta_\alpha)^2}\sin(\alpha + \theta - \theta_\alpha) \tag{19}$$

where

$$\alpha = \tan^{-1}\frac{z_r(\theta_\alpha)}{y_r(\theta_\alpha)} \tag{20}$$

as shown in Fig. 7 (c).

B. Method for Rollover prevention

The quadruped tracked mobile robot can prevent rollover by changing the angles of their legs. Initially, the legs take default posture to close fully as shown in Fig. 7 (a); the joint angles for the right leg are $\theta_1 = -90$[deg] and $\theta_2 = 135$[deg], and those for the left leg are $\theta_3 = 90$[deg] and $\theta_4 = -135$[deg]. When the rotation starts, the robot takes action for the rollover prevention. Fig. 8 shows the conceptual diagram of the action in the front view. In this case, the robot rotates in the right direction in the same way as Fig. 7.

In the state (a), as shown in the left panel of Fig. 8, the robot changes the joint angles of the left leg θ_3 and θ_4 to shift the position of mass center so that h_0 becomes minimum to the rotational angle θ. However, the Z value of tip position of the left leg must be positive.

After the tip of the right leg touches ground, the robot keeps touching the tip on the ground to increase h_{max} as shown in the right panel of Fig. 8. Fixing the joint-1 angle, θ_1, as -90 degrees, the joint-2 angle can be determined as

$$\theta_2 = \pi - \theta - \cos^{-1}(\frac{a}{l_{r2}}). \tag{21}$$

where l_{r2} is the length of the second link of the right leg, and

$$a = (z_{offset} + l_{r1})\cos\theta \tag{22}$$

where z_{offset} is the distance between the joint-1 and the rotation center on YZ plane.

C. Simulation

We computed S_{NE} to the joint angles at each rotation angle θ to determine appropriate posture of the leg so that S_{NE} becomes maximum.

Fig. 9 shows the result when $\theta = 15$[deg] in the rolling state (a). In this case, the left leg should be controlled to avoid touching ground for the right leg. Thus, S_{NE} to the joint angles of the left leg, θ_3 and θ_4, are plotted. In this result, the angles by which the left leg touches ground or interferes with any other part are excluded. The red point in Fig. 9 shows the maximum $S_{NE} = 256.65$[mm] at which $\theta_3 = -41$[deg] and $\theta_4 = 45$[deg]. This result indicates that the robot is able to stable as possible by controlling the joints of left leg to these angles to prevent rollover at the rotation angle $\theta = 15$[deg].

This computation was made for the other rotation angles in the rolling state (a) and appropriate θ_3 and θ_4 were obtained so that S_{NE} becomes maximum for each θ. Fig. 10 shows the result. The blue line shows θ_3 and red line shows θ_4.

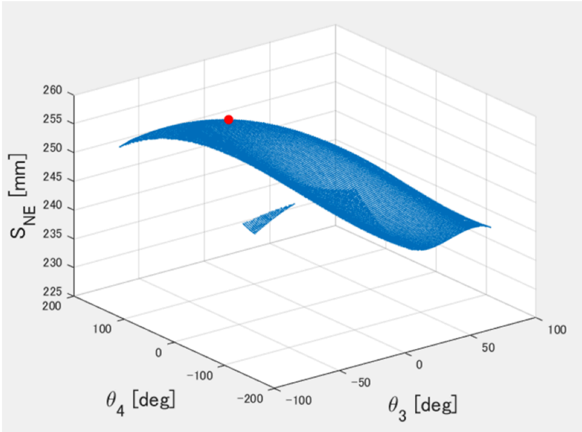

Fig. 9. S_{NE} to θ_3 and θ_4 at $\theta = 15$[deg]

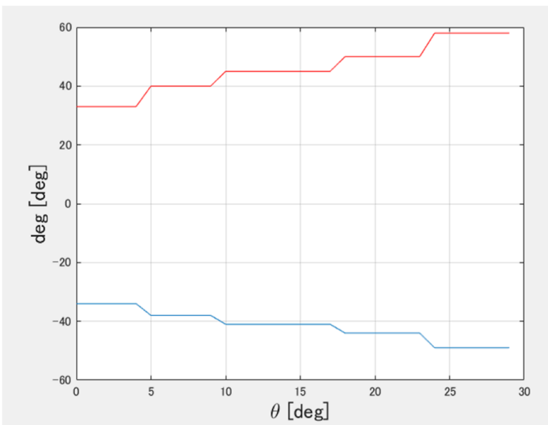

Fig. 10. Appropriate θ_3 (blue line) and θ_4 (red line) to θ

The result shows that the robot can be optimal posture by changing θ_3 and θ_4 in the range of -55 to -45 degrees and 20 to 60 degrees respectively.

We also computed S_{NE} in the rollover prevention action. Fig. 11 shows the change of S_{NE} to the rotation angles θ when the robot takes normal posture and acts rollover prevention.

The lower blue line in the figure shows the result in the default posture of the legs. The range of (a), (b), and (c) show the rolling states corresponding to them indicated in Fig. 7. When $\theta = 29$[deg], the tip of right leg touches the ground and the angle at that time becomes θ_α, then the radius of rotation changes. Therefore, after that, the decreasing rate of S_{NE} changes.

The upper line shows the S_{NE} when acting rollover prevention. In the range of (d), the joint angles of the legs, θ_3 and θ_4, that maximize S_{NE} were given based on the simulation results shown in Fig. 10. The result shows that S_{NE} can be increased about 30[mm] by the leg posture control in the rolling state (a).

The upper two gray (e) and red (f) lines show the S_{NE} when acting rollover prevention in the rolling state (c) after the tip of right leg touching the ground. The gray (e) line shows the result when the left leg keeps default posture. The most upper red (f) line shows the result when the joint angles of the left leg are fixed to be $\theta_3 = -49$[deg] and $\theta_4 = 58$[deg], which are the angles for the optimal posture at the rolling state (b). The result shows that the robot is still stable even when θ increases in both cases because h_{max} also increases as the tip position touching ground becomes further and the distance between the tip and mass center becomes longer; the amount of increase of h_{max} is large to the height of mass center. Specially, S_{NE} in the latter case (f) becomes higher because the mass center of the robot is lower.

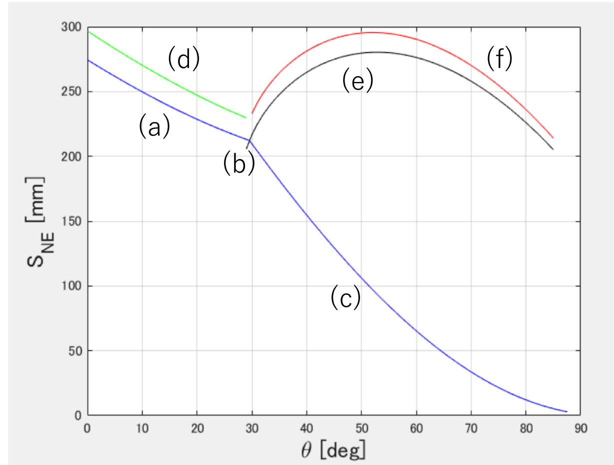

Fig. 11. S_{NE} to rotation angles θ

These results can be applied for the robot to control joint angles so that it prevents rollover properly to the rotation angle θ measured by the IMU.

VI. EXPERIMENTS

Based on the results of simulation described in Section V-C, several experiments for rollover prevention by actual robot were conducted. The robot controlled its joint angles of the leg to be optimal angles according to the rotation angles measured by the IMU sensor.

Fig. 12 shows an overview of the experiment. We used a sloped narrow straight long board to make rolling situation of the robot. The robot moved forward so that the left track only run on the board and the rotation angle increased as the robot running. The motion in the rolling state (a) were only experimented this time because the angle of the robot did not exceed 30[deg]. Fig. 12 indicates the view when the rotation angle $\theta = 20$[deg]. The appropriate joint angles at that state are $\theta_3 = -44$[deg] and $\theta_4 = 50$[deg] as shown in Fig. 10. The robot controlled the left leg to be such joint angles at that time.

As the result, the robot was able to controlled the left leg appropriately to maximize S_{NE} in the robot running.

In Fig. 13, the blue line shows the rotation angles measured by the IMU sensor in the running. The orange line shows the true values, which were computed from the robot

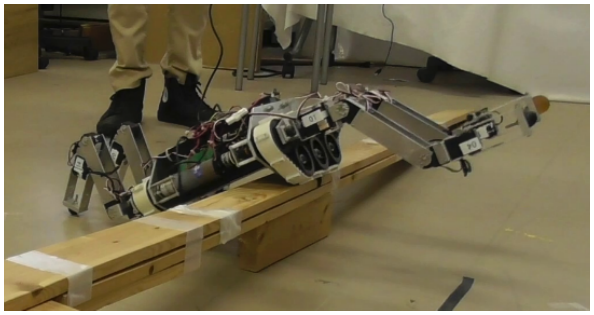

Fig. 12. Overview of rolling experiment ($\theta = 20$[deg])

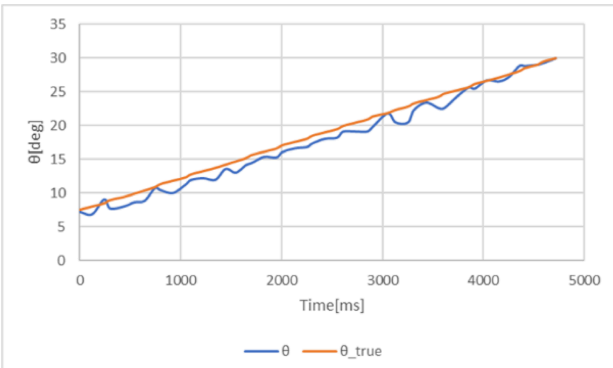

Fig. 13. Rotaion angles in the robot running

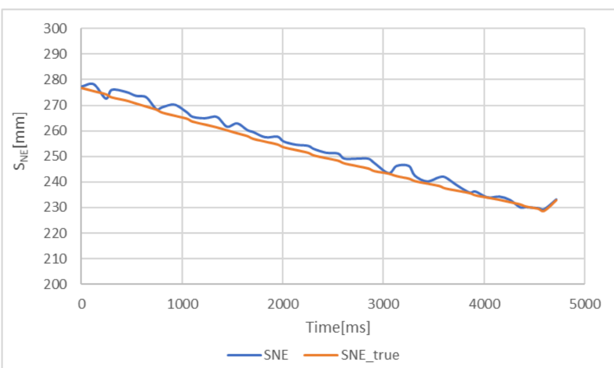

Fig. 14. S_{NE} in the robot running

position at each time in the experimental setup. This result shows that correct angles were measured in the experiment.

In Fig. 14, the blue line shows the computed S_{NE} by taking appropriate posture of the leg according to the measured rotation angles at each time in the robot running. The orange line shows the true values which were computed from the true rotation angles of Fig. 13 at each time. This result shows the valid control of the leg was performed to be most appropriated S_{NE} even though it decreased as the increase of the rotation angle in the running.

VII. CONCLUSIONS

This study presented the method for rollover prevention for the quadruped tracked mobile robot by controlling the joint angles of the legs. The fundamental experiments were conducted and their results confirmed that appropriate leg control is possible according to the rolling states. Specially, the quadruped tracked mobile robot has high potentials in moving and working. This rollover prevention considered in this study is also indicating high moving ability of the robot. However, using S_{NE}, static information is only considered for stability estimation in this study. Even though acceleration of translation and angular velocity were used for the computation of the center of rotation and rotation axis, these could not give valid information. As future work, dynamic information should be used properly as the index for stability evaluation. Also, the rollover prevention should be examined in more complex situation which occurs rotations around an axis of various directions.

REFERENCES

[1] T. Fujita and Y. Tsuchiya, “Development of a Quadruped Tracked Mobile Robot,” in *The ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE*, 2015.

[2] K. Ohno, E. Takeuchi, V. Chun, S. Tadokoro, T. Yuzawa, T. Yoshida, and E. Koyanagi, “Rollover avoidance using a stability margin for a tracked vehicle with sub-tracks,” in *2009 IEEE International Workshop on Safety, Security Rescue Robotics (SSRR 2009)*, 2009, pp. 1–6.

[3] K. Talke, L. Kelley, P. Longhini, and G. Catron, “Tip-over prevention through heuristic reactive behaviors for unmanned ground vehicles,” in *Proc. of SPIE Vol*, vol. 9084, pp. 90 840L–1.

[4] S. Hirose, Y. Fukuda, K. Yoneda, A. Nagakubo, H. Tsukagoshi, K. Arikawa, G. Endo, T. Doi, and R. Hodoshima, “Quadruped walking robots at tokyo institute of technology,” *IEEE Robotics and Automation Magazine*, vol. 16, no. 2, pp. 104–114, 2009.

[5] S. Hirose, H. Tsukagoshi, and K. Yoneda, “Normalized energy stability margin and its contour of walking vehicles on rough terrain,” in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 1, 2001, pp. 181–186.