Position Correction Using Elevation Map for Mobile Robot on Rough Terrain

Shintaro UCHIDA Shoichi MAEYAMA Akihisa OHYA Shin'ichi YUTA

Intelligent Robot Laboratory
University of Tsukuba
Tsukuba, 305-8573 JAPAN

E-mail : {maeyama,ohya,yuta}@roboken.esys.tsukuba.ac.jp
URL : http://www.roboken.esys.tsukuba.ac.jp/

Abstract

This paper proposes a new method of position correction for a mobile robot on rough terrain. The strong point of our method is that no external sensor is needed for position correction. The information about position correction is derived from comparing between the estimated position and elevation map of the terrain by using the knowledge that the robot must touch the ground. Some experimental results are also shown.

1 Introduction

The aim of this research is to develop a mobile robot which can safely navigate on rough terrain. It is assumed that the target environment of this research has smooth surface comparing with the size of the robot (see Figure 1). In order to navigate in such a terrain, the robot needs to keep knowing its position in three dimensions. The robot can estimate the position by dead reckoning using its internal sensors. But the robot occasionally has to cancel the cumulative errors of the dead reckoning. Several methods have already proposed using some external sensors to overcome this problem. For example, Aono[1] uses DGPS to cancel the cumulative errors of dead reckoning. However, no external sensor is needed if the robot is given an elevation map[2] of the environment in advance, because the error can be reduced by comparing the robots position/posture and the map information.

In this paper, we present the method for three dimensional position estimation, and propose a method of position correction using elevation map of terrain to cancel the cumulative errors of estimated position and posture which are obtained by dead reckoning.

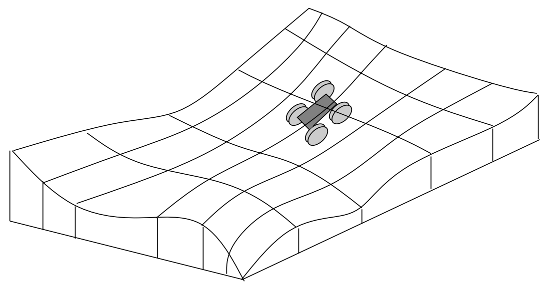

Figure 1: Image of target rough terrain environment which is rather smooth than the size of robot.

2 3D Position by Dead Reckoning

2.1 Update of 3D Position

The position and posture of the robot in 3D space at discrete time n can be represented by vector $\boldsymbol{p}_n$ as follows:

$$\boldsymbol{p}_n = (\ x_n \quad y_n \quad z_n \quad \alpha_n \quad \beta_n \quad \gamma_n\ )^T \tag{1}$$

x_n, y_n, z_n denote a robot position. $\alpha_n, \beta_n, \gamma_n$ denote yaw angle, pitch angle, roll angle of the robot respectively. The position and posture $\boldsymbol{p}_n$ is updated every sampling time period as follows:

$$\boldsymbol{p}_{n+1} = \boldsymbol{p}_n + \boldsymbol{A} \tag{2}$$

where,

$$\boldsymbol{A} = \begin{bmatrix} v_n \cos\beta_n \sin\alpha_n \Delta \\ v_n \cos\beta_n \cos\alpha_n \Delta \\ v_n \sin\beta_n \Delta \\ \frac{\omega_{yaw}+\dot{\beta}\sin\gamma}{\cos\beta_n \cos\gamma_n}\Delta \\ \dot{\beta}\Delta \\ \dot{\gamma}\Delta \end{bmatrix} \quad (3)$$

$$\begin{aligned} \dot{\beta} &= \frac{\beta_n - \beta_{n-1}}{\Delta} \\ \dot{\gamma} &= \frac{\gamma_n - \gamma_{n-1}}{\Delta} \end{aligned} \quad (4)$$

Δ denotes sampling time interval. v_n denotes robot velocity. $\boldsymbol{A}$ represents an incremental vector.

2.2 Prediction of Error Covariance

The robot keeps not only the position but its covariance matrix which indicates uncertainty of the estimated position. The MLE method for the position correction requires calculating covariance matrix of the robot's position. Covariance matrix Σ_p denotes the covariances of the position and the posture as follows.

$$\Sigma_p = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & \sigma_{x\alpha} & \sigma_{x\beta} & \sigma_{x\gamma} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{yz} & \sigma_{y\alpha} & \sigma_{y\beta} & \sigma_{y\gamma} \\ \sigma_{zx} & \sigma_{zy} & \sigma_z^2 & \sigma_{z\alpha} & \sigma_{z\beta} & \sigma_{z\gamma} \\ \sigma_{\alpha x} & \sigma_{\alpha y} & \sigma_{\alpha z} & \sigma_\alpha^2 & \sigma_{\alpha\beta} & \sigma_{\alpha\gamma} \\ \sigma_{\beta x} & \sigma_{\beta y} & \sigma_{\beta z} & \sigma_{\beta\alpha} & \sigma_\beta^2 & \sigma_{\beta\gamma} \\ \sigma_{\gamma x} & \sigma_{\gamma y} & \sigma_{\gamma z} & \sigma_{\gamma\alpha} & \sigma_{\gamma\beta} & \sigma_\gamma^2 \end{bmatrix} \quad (5)$$

The following equation successively gives the covariance matrix Σ_p every sampling time.

$$\Sigma_p(t+\Delta) = \boldsymbol{J}\Sigma_p(t)\boldsymbol{J}^T + \boldsymbol{K}\Sigma_m\boldsymbol{K}^T + \Delta^2\Sigma_N \quad (6)$$

Σ_N includes a covariance matrix of calculation and sampling error. $\boldsymbol{J}$ and $\boldsymbol{K}$ are the Jacobian matrices of $\boldsymbol{F}$ concerning $\boldsymbol{p}$ and $\boldsymbol{m}$, respectively. $\boldsymbol{F}$ is a vector of functions which expresses the 3D position update as follows:

$$\boldsymbol{F} = \boldsymbol{p}_n + \boldsymbol{A} \quad (7)$$

$\boldsymbol{m}$ is a vector which indicates elements of measurement as follows.

$$\boldsymbol{m} = (\ v \quad \omega_{yaw} \quad \dot{\beta} \quad \dot{\gamma}\ )^T \quad (8)$$

Covariance matrix Σ_m is the covariance of $\boldsymbol{m}$. It is assumed that each element of $\boldsymbol{m}$ has no correlation. And then Σ_m is expressed as follows:

$$\Sigma_m = \begin{bmatrix} \sigma_v^2 & 0 & 0 & 0 \\ 0 & \sigma_{\omega_{yaw}} & 0 & 0 \\ 0 & 0 & \sigma_{\dot{\beta}} & 0 \\ 0 & 0 & 0 & \sigma_{\dot{\gamma}} \end{bmatrix} \quad (9)$$

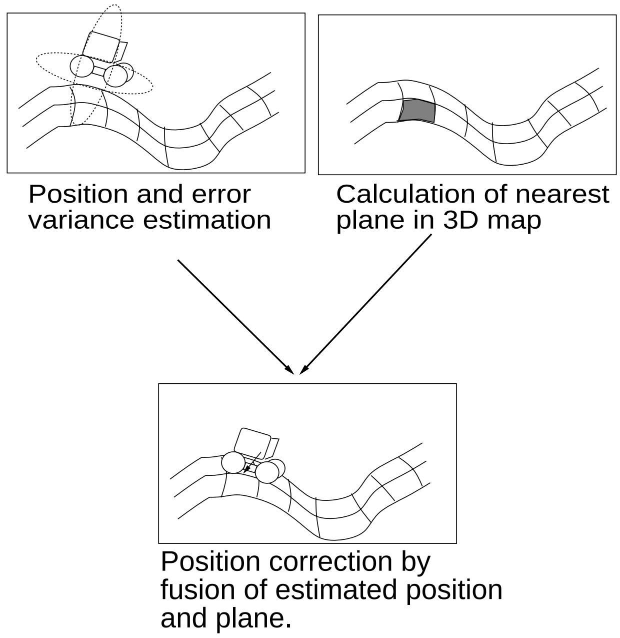

Figure 2: The way of position correction using 3D map.

3 Position Estimation Procedure by 3D Map

In the dead reckoning system the cumulative error increases as it is shown in the equation(6). We propose a method in which not external sensor but 3D map is used in order to cancel the error. Figure 2 shows the way of the correction. The corrected position is obtained by fusing the terrain equation which is derived from 3D map and estimated position by dead reckoning. The position of the robot is fused with the terrain equation using Maximum Likelihood Estimation(MLE) method. The position correction using MLE is based on the method of ours[3]. This time we propose a new method of three dimensional position correction using no external sensor, extending the method of two dimensional position correction using external sensors which was proposed in [3]. Figure 3 shows the flow of the position correction procedure. In the following sections, 3D map, approximated plane, and position correction are explained.

3.1 3D Map

It is assumed that the robot is given a 3D map of rough terrain in advance. The 3D map is expressed

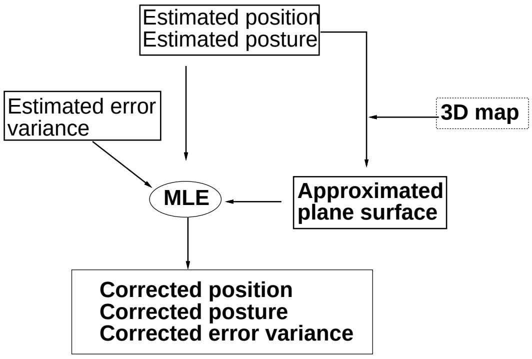

Figure 3: The flow of position correction by Maximum Likelihood Estimation(MLE).

by an elevation map[2] in this method. The elevation map has altitude information (z value) on each crossing point of x and y.

3.2 Approximated Plane Surface

In this study, it is assumed that the target environment has sufficient smooth surface comparing with the size of the robot. So, the elevation map can be approximated by a plane surface around robot's position. The plane surface is expressed by equation (10). The coefficients of the equation are derived from the nearest three points around the robot in the elevation map.

$$z = Ax + By + C \tag{10}$$

3.3 Correction of Error Covariance and Position and Posture

The robot always touches the ground by gravity. Therefore, the following equation $\boldsymbol{g}$ must be satisfied.

$$\boldsymbol{g}(\boldsymbol{p}, \boldsymbol{s}) = [Ax + By + C - z] = \mathbf{o} \tag{11}$$

where $\boldsymbol{s}$ is a coefficient vector of the plane which the robot exists on.

$$\boldsymbol{s} = \begin{bmatrix} A & B & C \end{bmatrix}^T \tag{12}$$

The covariance matrix $\Sigma_{p_{corrected}}$ indicates Σ_p which is corrected by the following equation.

$$\Sigma_{p_{corrected}} = (\Sigma_p^{-1} + \boldsymbol{U}_p^T \Sigma_{su}^{-1} \boldsymbol{U}_p)^{-1} \tag{13}$$

In this equation, $\boldsymbol{U}_p$ is obtained from normalized Jacobian vector $\boldsymbol{u}_p$ of $\boldsymbol{g}$ concerning $\boldsymbol{p}$ as follows:

$$\boldsymbol{u}_p = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{p}} = \begin{bmatrix} \frac{a}{|jp|} & \frac{b}{|jp|} & \frac{-1}{|jp|} & 0 & 0 & 0 \end{bmatrix} \tag{14}$$

where,

$$|jp| = \sqrt{a^2 + b^2 + 1} \tag{15}$$

To expand vector $\boldsymbol{u}_p$ to 6×6 matrix $\boldsymbol{U}_p$, five independent vectors are added to $\boldsymbol{u}_p$. $\boldsymbol{U}_p$ is

$$\boldsymbol{U}_p = \begin{bmatrix} \frac{a}{|jp|} & \frac{b}{|jp|} & \frac{-1}{|jp|} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

Σ_{su} is calculated from Σ_s through a coordinate transformation. Σ_s is the covariance matrix of $\boldsymbol{s}$.

$$\Sigma_s = \begin{bmatrix} \sigma_a^2 & 0 & 0 \\ 0 & \sigma_b^2 & 0 \\ 0 & 0 & \sigma_c^2 \end{bmatrix} \tag{17}$$

And Σ_{su}^{-1} can be represented as

$$\Sigma_{su}^{-1} = \begin{bmatrix} (J_s \Sigma_s J_s^T)^{-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{18}$$

where,

$$J_s = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{s}} \tag{19}$$

After calculating the corrected covariance matrix of position Σ_p, the corrected position $\boldsymbol{p}_{corrected}$ is derived from

$$\boldsymbol{p}_{corrected} = \boldsymbol{p} + \Sigma_{p_{corrected}} \boldsymbol{U}_p^T \Sigma_{su}^{-1} \boldsymbol{P}_{su} \tag{20}$$

where, $\boldsymbol{P}_{su}$ is an error vector of $\boldsymbol{g}(\boldsymbol{p}, \boldsymbol{s})$ which is normalized by $|jp|$ as follows:

$$\boldsymbol{P}_{su} = \left(\frac{-\boldsymbol{g}(\boldsymbol{p}, \boldsymbol{s})}{|jp|} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \right)^T \tag{21}$$

When the error of each parameter of position and posture has no correlation, the position and the posture are corrected to the direction of the vector $\boldsymbol{u}_p$. However, they could be also corrected on the other two axes which are independent of the vector $\boldsymbol{u}_p$ when the errors of them have some correlation.

4 Experiment

4.1 Experimental System

4.1.1 Mobile Base

Figure 4 is a photograph of the mobile robot which is designed for navigation on rough terrain. A radio

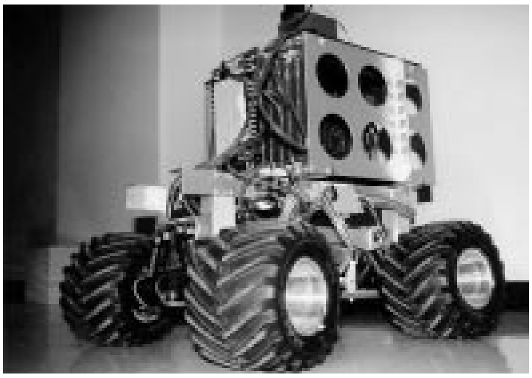

Figure 4: Mobile robot for rough terrain "AQURO".

Table 1: AQURO's specifications.

Way of wheel driving	Four wheels drive
Length of body	460mm
Width of body	420mm
Height of body	390mm
Diameter of tires	150mm
Weight(including buttery)	8kg
Voltage of power source	24V

controlled car (USA-1 NITRO CRUSHER, Kyosho) remodeled into the autonomous mobile robot. This robot is named "AQURO". AQURO is self-contained, namely, electrical motors, controllers, batteries are all mounted on the robot. The specifications of AQURO are shown in Table 1.

4.1.2 Sensors

Two encoders are mounted near the rear tires. The rear tire's angular velocities are measured to calculate the translational velocity v. The shafts of encoders are connected to rear tire's shafts by rubber belts and pulleys. A Fiber Optical Gyro is mounted on the top of the robot to measure yaw angle(Figure 5). The yaw angular velocity ω_{yaw} is calculated from the yaw angle. An inclinometer is also mounted on the top of the robot to measure pitch and roll angle(Figure 6). The pitch and roll angular velocity $\dot{\beta}, \dot{\gamma}$ are calculated from pitch and roll angle.

Figure 5: Fiber optical gyro (JG-108FD, JAE).

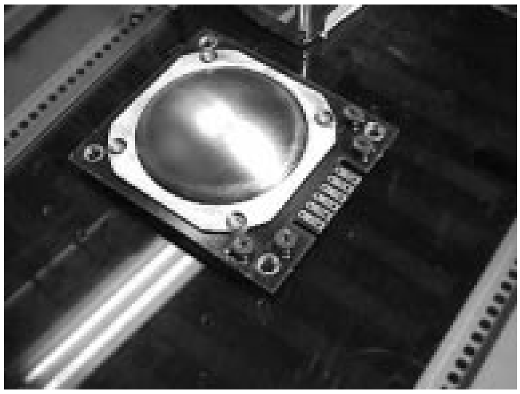

Figure 6: Inclinometer (DAS-20, Lucas).

4.1.3 Control System

Figure 7 shows the control system of AQURO. The control system is divided into three modules. Each module has its own CPU(T805). A master module decides the robot action according to the situation. A locomotion module controls motors and steering. The servo loop is executed every 5ms. The locomotion module receives a command from the master module. The commands are line trace, curved line trace, speed control, and so on. A position estimation module calculates the robot's position and the covariance matrix of the position every 40ms. And it also corrects the position using elevation map by the proposed method.

One of the special features of this control system is the way of using position estimation module. The position estimation module automatically correct the robot position. So the user don't have to care about

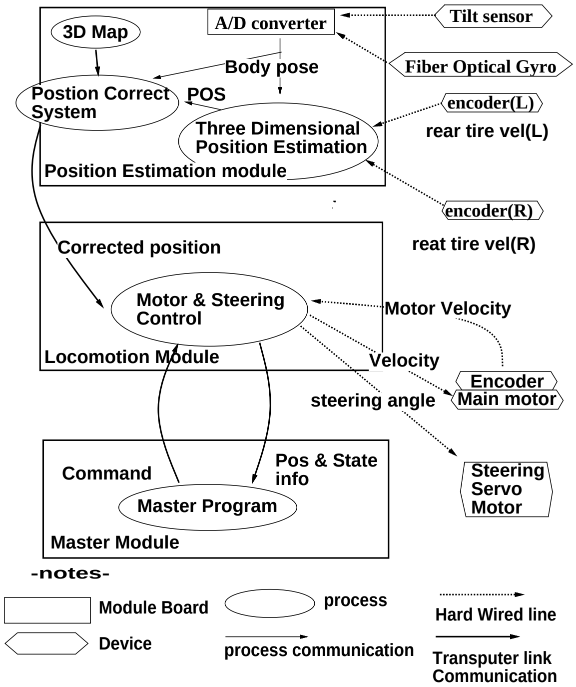

Figure 7: Control system of AQURO.

the correction of the position. All the user have to do is making program of the robot's behavior just for master module.

4.2 Experimental Terrain

The experiments of position correction are done in an artificial environment. The terrain was made of one expanded polystyrene block and two wooden boards. The elevation map which robot has are shown in Figure 8.

4.3 Experimental Condition of Error Variance

The experimental conditions of error variances are shown in Table 2 and Table 3. The initial error variances mean the initial errors of position and posture when the robot is on start point. The measurement error variances mean the errors which arose from sensing. The values of variances are determined empirically.

4.4 Experimental result

The velocity of mobile robot was 20(mm/s). Figure 9 shows x, z coordinates of position estimation and

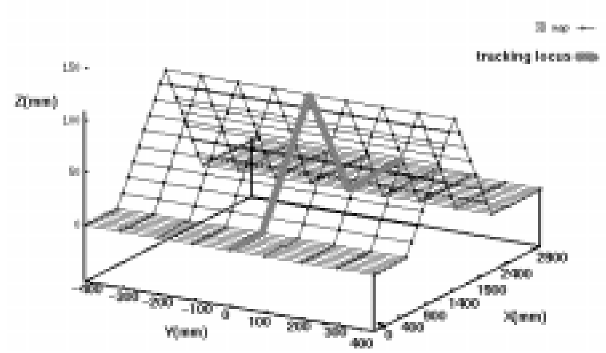

Figure 8: The 3D map of terrain and tracking locus.

Table 2: Initial error variances in the experiments.

$\sigma_x, \sigma_y, \sigma_z$	60(mm)
$\sigma_\alpha, \sigma_\beta, \sigma_\gamma$	10(deg)

correction result. The correction was executed every 4 seconds. We can see that the error is reduced by the position correction. The direction and the amount of position correction depends on

1. The direction of the normal vector of map plane,
2. The estimated error variances.

In the experiment the y elements of the plane normal vectors are 0. So the position was corrected in only z and x direction.

Figure 10 shows the result of position estimation and error ellipse with position correction. Figure 11 shows the error ellipse without position correction. The error ellipse means the possibility of existence area of robots. The error ellipse is calculated from Σ_p.

Table 3: Measurement error variances in the experiments.

σ_v	10(mm/s)
$\sigma_{\omega_{yaw}}$	0.1(deg/s)
$\sigma_\beta, \sigma_\gamma$	0.1(deg)
σ_a, σ_b	0.1
σ_c	5

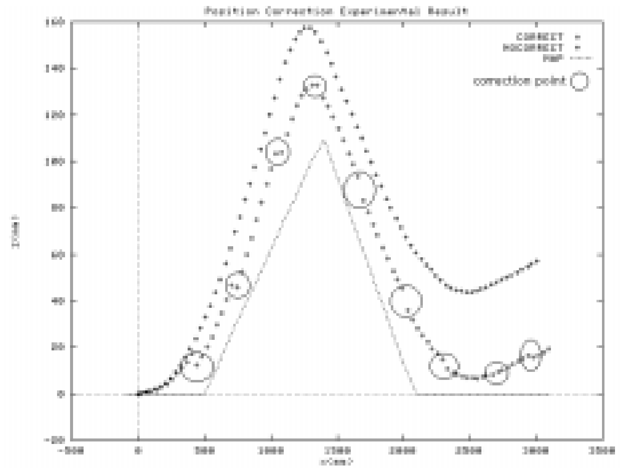

Figure 9: Map and estimated position with correction, without correction(x, z coordinates). This figure shows that positions of z is corrected. The correction is executed every 4 seconds. (Notes: the magnification of x, z coordinates are very different.)

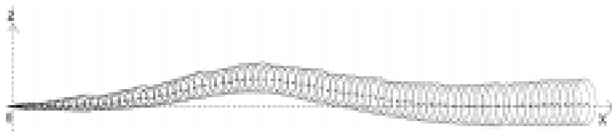

Figure 10: Estimated position and error variance with correction. The error variance of z is corrected, and is kept small. The correction is executed every 4 seconds. (Notes: the magnification of z coordinate is around 1.5 times as long as x coordinate.)

The larger eigenvalue of Σ_p is the major axis length of the error ellipse. The direction of eigenvector is the direction of the major axis of the error ellipse. The size of ellipse is kept small with correction.

5 Conclusions

In this paper we proposed a new method of position correction using no external sensors but only 3D map. The robot uses 3D map and the knowledge that the robot always touches the ground. The effectiveness of this method was shown by the experiment using a real mobile robot.

In some terrain condition, 3D position correction using this method may be limited. Some other external sensor needs to be used in this case. The problem of how to obtain an elevation map should be considered in future.

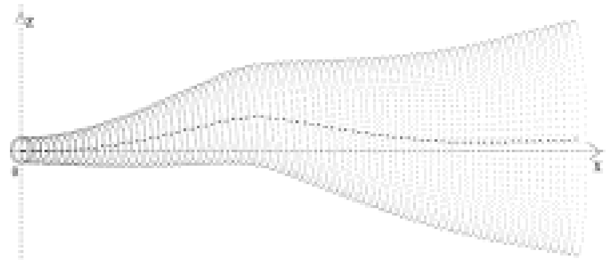

Figure 11: Estimated position and error variance without correction. The error variance of z is not corrected, and doesn't become small. (Notes: the magnification of z coordinates is around 1.5 times as long as x coordinate.)

References

[1] T. Aono et al.: "Position Estimation using GPS and Dead Reckoning", Proceeding of IEEE/SICE/RSJ International Conference Multisensor Fusion and Integration for Intelligent System (1996)

[2] R. Chatila et al.: "Autonomous Navigation in Natural Environments", Lecture Notes in Control and Information Sciences 200, Springer-Verlag (1993)

[3] Y. Watanabe and S. Yuta: "Position Estimation of Mobile Robots with Internal and External Sensors Using Uncertainty Evolution Technique", Proceedings of IEEE International Conference on Robotics and Automation, pp.2011-2016 (1990)

[4] Y. Fuke and E. Krotokv: "Dead Reckoning for a Lunar Rover on Uneven Terrain", Proceeding of IEEE International Conference on Robotics and Automation (1996)