Robust Path Planning against Pose Errors for Mobile Robots in Rough Terrain

Yuki Doi[1], Yonghoon Ji[1], Yusuke Tamura[1],
Yuki Ikeda[2], Atsushi Umemura[2], Yoshiharu Kaneshima[2], Hiroki Murakami[2],
Atsushi Yamashita[1], and Hajime Asama[1]

[1] The University of Tokyo
[2] IHI Corporation

Abstract. We propose a novel path planning method considering pose errors for off-road mobile robots based on 3D terrain map information. Mobile robots navigating on rough terrain cannot follow a planned path perfectly because of uncertainties such as pose errors. In this work, we represent such pose errors as error ellipsoids to use on collision check with obstacles in a map. The error ellipsoids are estimated based on extended Kalman filter (EKF) that integrates motion errors and global positioning systems (GPS) observation errors. Simulation and experiment results show that the proposed method enables mobile robots to generate a robust path against pose errors in a large-scale rough terrain map.

Keywords: path planning, rough terrain, random sampling, extended Kalman filter, error ellipsoid

1 Introduction

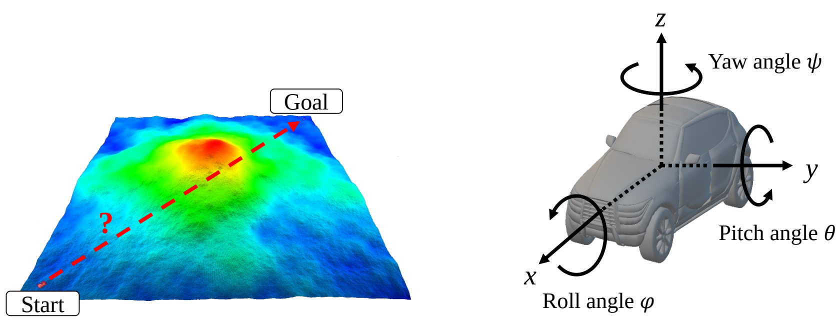
Disaster response activities are important to save human lives and resources. In recent years, mobile robots equipped with sensors such as a camera are used when disasters occur. Mobile robots are usually operated by a remote control when human cannot enter environments that are damaged by disasters. However, a remote control has limitations because an operator is hard to recognize a surrounding environment with information through a camera. Therefore, autonomous mobile robots that are able to plan a safe path avoiding dangerous areas are needed. In 2004 and 2005, DARPA (Defense Advanced Research Projects Agency) held DARPA ground challenge [1]. In this challenge, a lot of mobile robots tried to run a long way of 150 miles in rough terrain and five robots finished. These robots were provided the navigation path in advance. In case of a disaster situation, feasible path for mobile robots cannot be provided before a disaster occurs and given that we can only get very limited environment information. Hence, the path planning method which generates safe paths for mobile robots in rough terrains is very important.
Kuwata et al. proposed a real time motion planning method for autonomous cars [2]. They classified environment maps according to a risk of collision. Moreover, they reduced calculation time by preserving a former result and using it

during a next motion planning. Richter et al. proposed a path planning method for autonomous cars with a self-position estimation [3]. They set high risk allowances in areas where the cars did not construct environment maps in order not to plan paths in uncertain areas. Nevertheless, these methods mentioned above assumed 2D environments that cannot deal with rough terrain. In this respect, Ji et al. proposed the broad path planning method on a 3D environment map for mobile robots traveling rough terrain [4]. In this study, a motion model of a mobile robot was considered; hence, it was possible to generate feasible paths for navigation. Moreover, they avoided the problem of generating a path that could cause the robots to fall down by restricting robot angles of a inclination and radii of a gyration during the path planning. However, robot pose errors that should be managed for actual operation of the robot were not considered.
Generally, mobile robots are affected by pose errors in many respects while navigating in the real environment. Thus, the robot cannot follow the generated path perfectly due to the pose errors. If the generated path is close to obstacles, the robot may collide with the obstacles. To take the pose errors into account during path planning, van den Berg et al. proposed path a planning method that considers motion uncertainty and imperfect sensors [5]. They calculated probability of collision with generated paths and map to judge validations of paths. However, they also assumed 2D environments. Blackmore et al. and Lee et.al. proposed optimal robust path planning methods that use the random sampling integrated with chance constraints [6, 7]. Chance constraints enable to restrict probabilities of collision with obstacles by satisfying constraint equations. However, chance constraints can be calculated only when obstacles are convex polyhedrons and they assumed only 2D environments. There are many obstacles of various shapes in rough terrain; thus, this method cannot be applied when disaster occurs.
In this study, we propose a safe path planning method for mobile robots in a rough terrain when pose errors affect robots. The remainder of this paper is organized as follows. Sections Ⅱ introduces the problem definition and approach of the proposed robust path planning. Then, the method of an error estimation is presented in Section Ⅲ. Section Ⅳ introduces the sampling method considering the robot acceleration. The validity of the proposed path planning method is evaluated with the simulation and real experimental results in Section V. Finally, Section Ⅵ gives conclusions of this paper.

2 Approach

In this study, we propose a novel path planning scheme that can manage pose errors under the assumption that a four-wheeled vehicle as a mobile robot has a map of an entire environment consisting of 3D point cloud data in advance. An environment map is measured by unmanned aerial vehicles (UAV) in advance.

Figure 1(a) shows an example of a 3D point cloud of a rough terrain. This robot is expressed in six dimensional configurations, position information (x, y, z) and orientation angles (ϕ, θ, ψ) around axes, as shown in Fig. 1(b). Moreover, acceleration are restricted so as not to make movements that robots cannot



(a) 3D pointed cloud of rough terrain **(b)** Six dimensional configuration

Fig. 1: Assumptions of proposed path planning method.

handle. Maximum inclination angles of slopes are also restricted according to robot velocities; hence paths must select routes with lower angle than maximums. We also assume that the robot is equipped with two global positioning systems (GPS); hence, it can obtain its position and orientation information. In our previous work [4], we assumed that GPS sensors obtain true values of robot poses and a mobile robot can follow generated path, perfectly. However, as mentioned in Section 1, the robot is generally affected by observation errors of GPS and motion errors in real environments. In this study, we regard these two errors as pose errors. By taking the pose errors into consideration, the generated path produced by our proposed scheme ensures safety even if the mobile deviates from the planed path.

We use the random sampling algorithm [8] in the path planning method. The random sampling-based method cannot find optimal path; however, it can explore a map of large environment, quickly. The random sampling algorithm performs a path planning by generating robot configurations as new nodes and connecting them to existing ones. In the proposed method, robot velocities and angular velocities are sampled and nodes corresponding to robot poses are generated using a robot motion model. We defined paths avoiding collisions with obstacles even pose errors affect as robust paths. To consider effects of pose errors during a path planning, we adopted to estimate pose errors after each node are generated and to converted pose errors to error ellipsoids. We assumed average of GPS data equal generated node because mobile robots correct its position by the local control. Figure 2 shows the flowchart of our random sampling algorithm. In a conventional random sampling method, generated nodes are judged without the error estimation. On the other hand, in our random sampling algorithm, we estimate motion errors and observation errors by converting them into one error ellipsoid after generating each node. Error ellipsoids are used to judge validation of nodes by checking collisions with a 3D environment map. We also propose a new sampling method of robot velocities. This method enables to plan safe paths that mobile robots can follow.

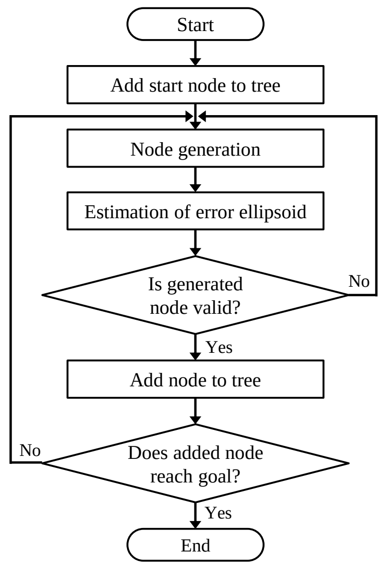


Fig. 2: Flowchart of proposed random sampling algorithm.

3 Error estimation

3.1 Pose error estimation of each node

The mobile robot generally obtains pose information from GPS and controlled by control input data to moves to the target position. In our random sampling algorithm, nodes corresponding to robot poses are generated assuming control inputs data and connected to existing ones. Within components of nodes, the robot position (x_t, y_t) and the yaw angle ψ_t are sampled by using the motion model. The position z_t, the roll angleϕ_t and pitch angle θ_t are determined from map information. When control input data $\mathbf{u}_t = (v_t, \omega_t)$ is given, the robot pose at time t $\mathbf{x}_t = (x_t, y_t, \psi_t)$ is given as the following motion model.

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1},\ \mathbf{u}_t) = \begin{pmatrix} x_{t-1} + v_t \mathit{\Delta} t \cos\left(\psi_{t-1} + \frac{\omega_t \mathit{\Delta} t}{2}\right) \\ y_{t-1} + v_t \mathit{\Delta} t \sin\left(\psi_{t-1} + \frac{\omega_t \mathit{\Delta} t}{2}\right) \\ \psi_{t-1} + \omega_t \mathit{\Delta} t \end{pmatrix}. \tag{1}$$

However, the mobile robot cannot perform the expected operation because of the pose errors consisting of observation errors and motion errors. Therefore, estimating pose errors is important to plan safe path for mobile robots. The observation errors occur when the mobile robot obtains pose information from the GPS and the motion errors occur when the mobile robot moves based on the control input data. We integrate these motion errors and observation errors into pose errors represented by an error ellipse. We estimate pose errors by using extended Kalman filter (EKF) [9] during path planning because it can integrate errors with different characteristics. EKF process is divided into two steps and the motion error and observation error are calculated in each step.

EKF prediction step In prediction step, a robot motion error is estimated. The motion error is determined by using Jacobian $\mathbf{J}$ derived from the motion model of the mobile robot $\mathbf{f}$. The Jacobians $\mathbf{J_x}$ corresponding the robot pose and $\mathbf{J_u}$ corresponding to the control input are given as follows:

$$\mathbf{J_x} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}, \tag{2}$$

$$\mathbf{J_u} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}. \tag{3}$$

The motion error at time t is expressed as the covariance matrix as follows:

$$\bar{\mathbf{\Sigma}}_{\mathbf{x}t} = \mathbf{J}_{\mathbf{x}t}\mathbf{\Sigma}_{\mathbf{x}t-1}{\mathbf{J}_{\mathbf{x}t}}^T + \mathbf{J}_{\mathbf{u}t}\mathbf{\Sigma}_{\mathbf{u}t}{\mathbf{J}_{\mathbf{u}t}}^T, \tag{4}$$

where $\bar{\mathbf{\Sigma}}_{\mathbf{x}t}$ is robot covariance matrix after applying motion error and $\mathbf{\Sigma}_{\mathbf{u}t}$ is input covariance. Each $\mathbf{\Sigma}_{\mathbf{u}t}$ is determined based on a robot position and a satellite arrangement. $\mathbf{\Sigma}_{\mathbf{x}t-1}$ denotes covariance matrix corresponding to the pose error at previous time step t-1.

EKF update step In this step, the covariance matrix $\bar{\mathbf{\Sigma}}_{\mathbf{x}}$ calculated in prediction step is updated by using GPS observation errors. The updated covariance matrix $\mathbf{\Sigma}_{\mathbf{x}}$ is given as follows:

$$\mathbf{S}_t = \mathbf{H}_t\bar{\mathbf{\Sigma}}_{\mathbf{x}t}(\mathbf{H}_t)^T + \mathbf{Q}_t, \tag{5}$$

$$\mathbf{K}_t = \bar{\mathbf{\Sigma}}_{\mathbf{x}t}(\mathbf{H}_t)^T(\mathbf{S}_t)^{-1}, \tag{6}$$

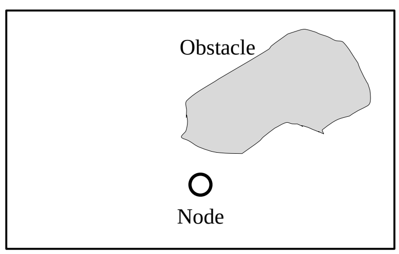
$$\mathbf{\Sigma}_{\mathbf{x}t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\bar{\mathbf{\Sigma}}_{\mathbf{x}t}. \tag{7}$$

Here, $\mathbf{Q}$ is as covariance matrix of the observation error and $\mathbf{H}$ is Jacobian of GPS observation. $\mathbf{H}$ is matrix which defines relationship between the observed value and the robot pose. In this case, observed values represent robot pose directly; thus, $\mathbf{H}$ is same to the identity matrix. $\mathbf{S}$ is observation uncertainty and $\mathbf{K}$ is Kalman gain. During the path planning, we use $\mathbf{\Sigma}_{\mathbf{x}t}$ in collision check and it is described in detail in next sebsection.

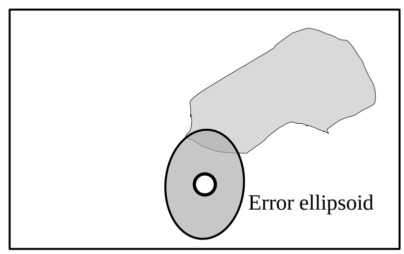
3.2 Collision check using error ellipsoid

The random sampling method judge validations of generated nodes by checking collisions with a map and angle values of a robot inclination. Our previous study

conducted collision checks between a map and a robot model with the fixed size, as shown in Fig. 3(a) [4]. Therefore, this method often generated nodes that were close to obstacles. In order to solve this problem, we propose the method that converts estimated pose errors to error ellipsoids and uses them in collision check. Figure 3(b) shows our collision check of the generated node during the random sampling. The proposed method checks collision with error ellipsoids and 3D object, directly. Error ellipsoids changes their sizes feasibly based on pose errors of a robot. Therefore, the proposed method can delete dangerous nodes that may collide because of pose errors. In the collision check, we use a flexible collision library (FCL) [10], the C++ library of collision detection.
Figure 4 shows the concept of our proposed random sampling scheme. In gray areas, GPS signals are not available and robots cannot identify their own poses. Figure 4(a) shows a conventional random sampling method checking collisions with maps and robot models fixed their sizes. Thus, this method often plans paths that are close to obstacles. On the other hand, we can plan a path considering the observation error and the motion error of mobile robots all at once. Hence, for example, safer path planning can be performed depending on different motion errors and a GPS situation, as shown in Fig. 4(b).

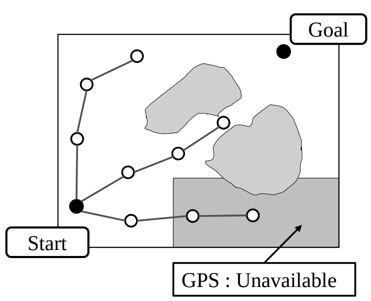


(a) Previous method that uses fixed robot size [4]

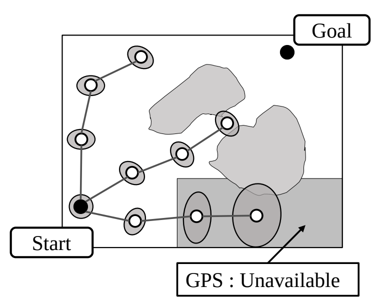


(b) Proposed method that uses error ellipsoid

Fig. 3: Collision check of generated node.



(a) Conventional random sampling method



(b) Proposed random sampling method

Fig. 4: Random sampling using error ellipsoids.

4 Sampling method of velocity

In this section, we describe the novel sampling method of robot velocities (i.e., control input data at node sampling). Our method samples the robot velocities and generate nodes that correspond to robot poses using the motion model. In the previous method [4], velocities are determined based on fixed probability distribution. They did not consider accelerations; hence, it often samples velocities exceeding the limit. Figure 5 shows the proposed sampling method of robot velocities. We determine a position of probability distribution of velocity based on the former one velocity value and determine a range based on the robot acceleration as shown in Fig. 5(a). Probability distribution of each sampling is shown in Fig. 5(b). It is mixed with normal distribution and uniform distribution. This method enables to restrict accelerations and plan paths that robots can follow.

5 Simulation

5.1 Simulation setting

In order to varify the effectiveness of the proposed path planner, we conducted simulations. A desktop computer with Intel core i7-6700 (3.40 GHz) CPU and 16.0 GB RAM memory was used to execute the proposed path planner. We used two maps in our simulation experiments The size of the map 1 that has rough terrain was 200 m^2. There were a 5 m narrow route to the left and a 12 m broad route to the right. The size of the map 2 was 100 m^2. There was a tunnel with a height of 7 m and a length of 30 m to the left. We assumed that the mobile robot was not able to get GPS information in the tunnel. We used path-directed subdivision tree (PDST) as a random sampling planner that determines how nodes are generated [11]. The robot was not able to follow if its acceleration was greater than 2.78 m/s^2. Moreover, we assumed that the standard deviation of the GPS observation was 1.0 m in position and 1.0 deg in orientation. Velocity errors of 10 % occurred. In addition, we also conducted the simulation with small velocity errors of 1 % in map 2. The sizes of the error ellipsoids were determined so that the mobile robot existed in ellipsoids with a probability of 95 %.

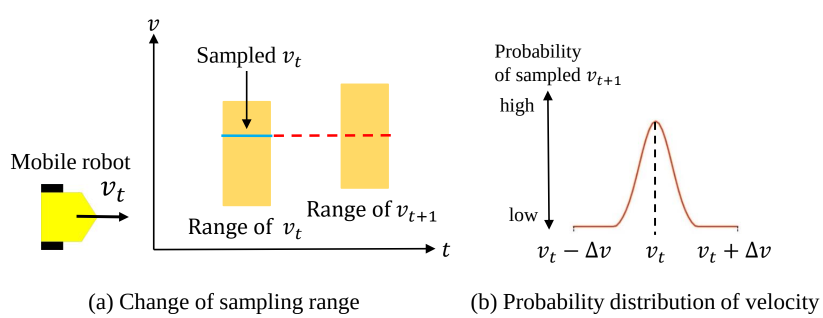
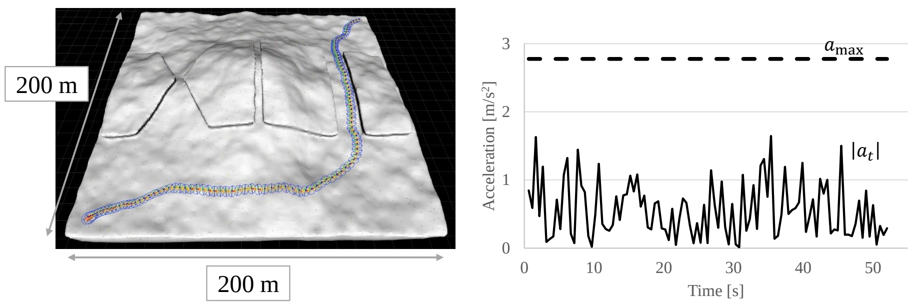


Fig. 5: Proposed sampling method.

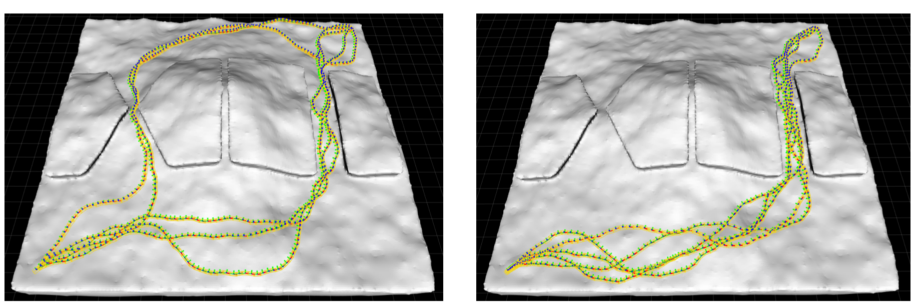
5.2 Simulation result

Figure 6(a) shows the generated path using the proposed method on map 1. The generated path is expressed in yellow line and error ellipsoids are expressed in blue lines. The proposed method was able to plan the safe path that error ellipsoid did not collide with the obstacles in the rough terrain. Further, the acceleration of the mobile robot kept within the limit as shown in Fig. 6(b). Figure 7 shows generated paths through five repeated simulations on map 1. The proposed methods generated each paths in less than five seconds. The previous method planed two paths that passed through a dangerous narrow route. On the other hand, all five paths generated by the proposed method avoided the dangerous. Moreover, the proposed method did not generate detour paths.
Figure 8 shows the difference of generated paths depending on motion error values on map 2. When the set motion error was small, it planed the path that passed through a tunnel because pose errors did not accumulate so much in the tunnel. On the other hand, when the set motion error was large, it planed the path that avoid a tunnel because the path planning method was not able to find a path avoiding collision in the tunnel. Figure 9 shows the difference in area where



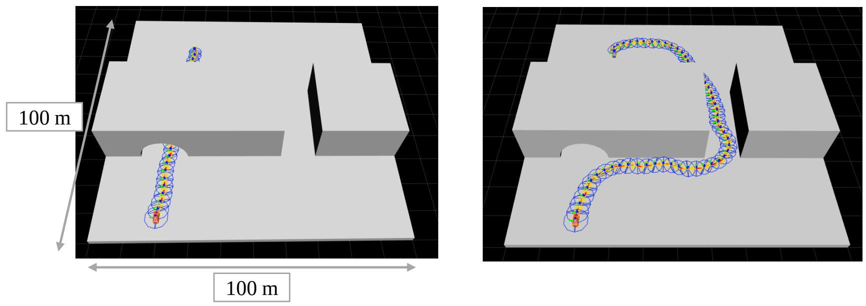
(a) Generated path **(b)** Change of acceleration

Fig. 6: Generated path by using proposed method.



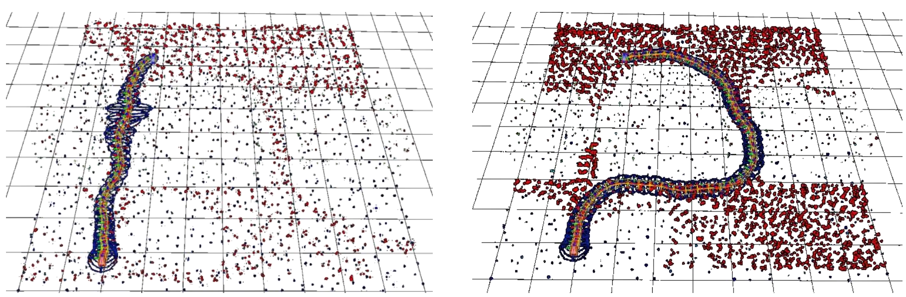
(a) Previous method[4] **(b)** Proposed method

Fig. 7: Generated paths through five repeated simulations on map 1.



(a) Result for velocity errors of 1 % **(b)** Result for velocity errors of 10 %

Fig. 8: Difference of generated paths depending on motion error values on map 2.



(a) Nodes for velocity errors of 1 % (b) Nodes for velocity errors of 10 %

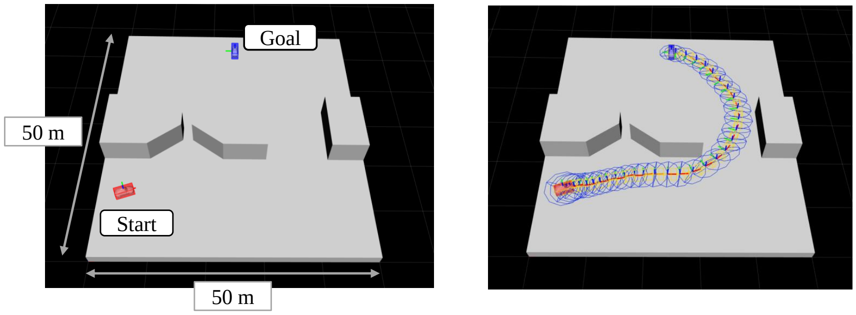
Fig. 9: Difference of generated nodes depending on motion error values on map 2.

nodes are generated on map 2. Here, nodes generated during path planning are expressed in red points. Nodes generated during path planning are expressed in red points. When the mobile robot cannot get GPS information, EKF cannot conduct the update step. For this reason, the motion error accumulated in the tunnel. Error ellipsoids will be larger than the tunnel when the motion error is large; hence, the proposed method did not generated nodes in the tunnel. However, error ellipsoids will not become very large when the motion error is small; hence, it generated nodes there. Therefore, the proposed method planed feasible paths that suitable for mobile robots errors.

6 Experiment

6.1 Experiment setting

Figure 10 shows path planning result for the field experiment. The size of the map was 50 m^2. There were a 2 m narrow route to the left and a 12 m broad route to the right. As shown in Fig. 10(b)., the proposed method generated a safe



(a) Field environment model

(b) Generated path

Fig. 10: Path planning for field experiment.

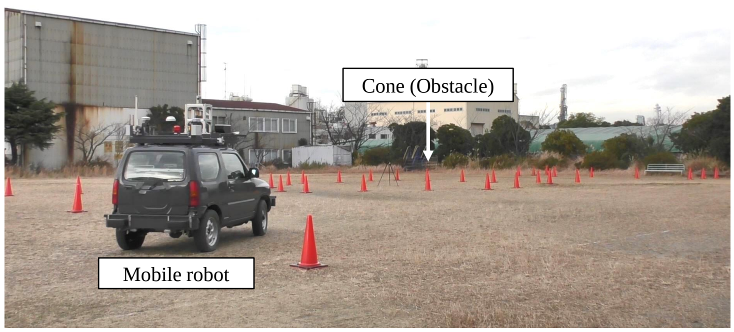
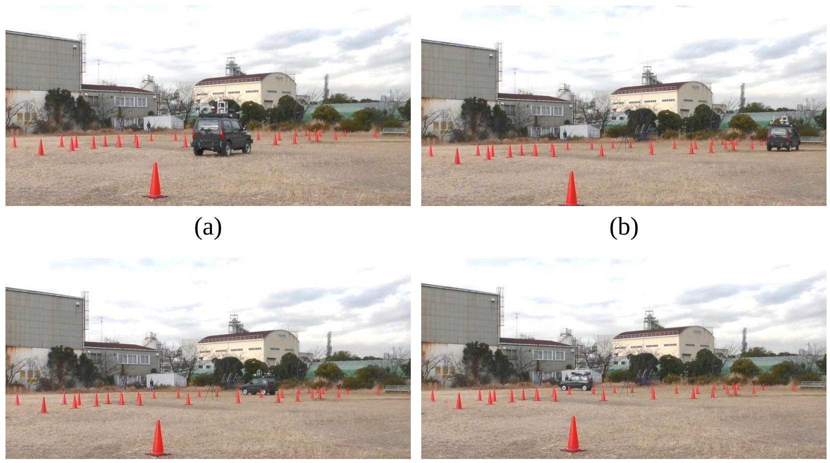


Fig. 11: Overview of field experiment.

path that passes through the broad route. The experimental environment and the mobile robot used for the field experiment are shown in Fig. 11. In order to install obstacles on the map as shown in Fig. 10, we used cones as obstacles in the field. The mobile robot was equipped with two GPS sensors; thus, it can obtain pose information. We assumed that observation errors of GPS followed a normal distribution and its standard deviations were 1.0 m in position and 1.0 deg in angle Moreover, we assumed that standard deviations of motion errors set at 10 % of robot input velocity.

6.2 Experiment result

Figure 12 shows the result of field experiment. The mobile robot traveled in order of (a) to (d). The robot was able to travel safely along the path generated in Fig. 10(b). Thus, we confirmed that the proposed path planning method is able to plan safe paths in real environments.



(a) (b) (c) (d)

Fig. 12: Result of field experiment.

7 Conclusion

We proposed the robust path planning method against pose errors for mobile robots. In the proposed method, we used EKF to estimate pose errors and checked collisions with error ellipsoid. This enables to find feasible paths taking the pose errors into consideration. Moreover, the proposed velocity sampling method can generate the paths based on the acceleration that the actual mobile robots can deal with.
The future work related to this study is to conduct online path planning. In this study, we assumed that the map information were given in advance. However, real terrain information may change while mobile robots are navigating. Therefore, in case of real applications, it is necessary to propose an effective method which processes surrounding environment maps and generate feasible path simultaneously.

Acknowledgement

This work was in part funded by ImPACT Program of Council for Science, Technology and Innovation (Cabinet Office, Government of Japan).

References

1. DARPA (Defense Advanced Research Projects Agency): Grand Challenge 2005 Report to Congress. http://archive.darpa.mil/grandchallenge/docs/grand_challenge_2005_report_to_congress.pdf, 2006.

2. Kuwata Y.,Teo J., Fiore G., Karaman S., Frazzoli E. and How J P.: Real-Time Motion Planning with Applications to Autonomous Urban Driving. IEEE Transactions on Control System Technology, Vol. 15, pp. 1105–1118, 2009.
3. Richter C., Ware J. and Roy N.: High-Speed Autonomous Navigation of Unknown Environments Using Learned Probabilities of Collision. Proceedings of the 2014 IEEE International Conference on Robotics and Automation, pp. 6114–6121, 2014.
4. Ji Y., Tanaka Y., Tamura Y., Kimura M., Umemura A., Kanashima Y., Murakami H., Yamashita A. and Asama H.: Adaptive Motion Planning Based on Vehicle Characteristics and Regulations for Off-Road UGVs. IEEE Transactions on Industrial Informatics. (under review).
5. van den Berg J., Abbeel P. and Goldberg K.: LQG-MP: Optimized Path Planning for Robots with Motion Uncertainty and Imperfect State Information. the International Journal of Robotics Research, Vol. 30, Issue 7, pp. 895–913, 2011.
6. Blackmore L., Li H. and Williams B.: A Probabilistic Approach to Optimal Robust Path Planning with Obstacles. Proceedings of the American Control Conference, pp.7–13, 2011.
7. Lee S. U. and Iagnemma K.: Robust Motion Planning Methodology for Autonomous Tracked Vehicles in Rough Environment Using Online Slip Estimation. Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3589–3594, 2016.
8. LaValle S. M.: Rapidly-Exploring Random Trees: A New Tool for Path Planning. Computer Science Department, Iowa State University, Technical Report TR 98-11, pp. 1–4, 1998.
9. Greg W. and Gary B.: An Introduction to the Kalman Filter. Proceedings of ACM SIGGRAPH, Course 8, 2001.
10. Pan J., Chitta S. and Manocha D.: FCL: A General Purpose Library for Collision and Proximity Queries. Proceedings of the 2012 IEEE International Conference on Robotics and Automation, pp.3859–3866, 2012.
11. Ladd A. M. and Kavraki L. E.: Fast Tree-Based Exploration of State Space for Robots with Dynamics. Algorithmic Foundations of Robotics VI, pp. 297–312, 2004.