The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Motion Planning of an Autonomous Mobile Robot Considering Regions with Velocity Constraint

Kiyohiro Goto, Kazuyuki Kon and Fumitoshi Matsuno

***Abstract*— Recently various autonomous mobile robots are developed for practical use. For coexistence with the robots and human in the real environment, the consideration of safety is very important. We should consider a region with a limitation of a maximum velocity of a mobile robot for the safety. In this paper, we propose path planning and trajectory generation methods for a mobile robot which moves in the environment with predetermined velocity constraints. In order to demonstrate the validity of the proposed methods, numerical simulations and experiments have been carried out.**

I. Introduction

In recent years various autonomous mobile robots are developed for practical use, such as service robots[1], [2], rescue robots[3] and electric wheelchairs[4], etc. For such autonomous robots operating in the real environment, "safety" is one of the most important factors. However, there are many hazardous regions in the real environment for autonomous locomotion of the robot. In such hazardous regions, it is difficult to recognize and judge whether the environment is enough safe to move at fast speed, only from the equipped sensory data.

In order to improve the safety of the autonomous locomotion, we propose a concept named "*The Region with Velocity Constraint*" (RVC). The RVCs are the regions where the velocity of the robot is constrained. The velocity constraint in each RVC is registered by an operator in advance, based on a priori map information. In other words, by registering the hazard regions as the RVCs, the predicted hazard can be avoided. Moreover, we also proposed a new motion planning method which enables the autonomous locomotion in the environment including the RVCs. The proposed motion planning method is divided into two parts: the first part is path planning method; and the second part is trajectory generation method. Note that, the environment addressed in this study is only static and known in advance.

There are a lot of works in the area of the motion planning for an autonomous robot, path planning (e.g. [5], [6]), trajectory generation (e.g. [7]-[9]), and obstacle avoidance (e.g. [10], [11]), etc. These typically represent the environment as a binary map, '0' indicates a free-space and '1' indicates an obstacle. Since the surroundings of the obstacles are also represented as free space, the existing methods allow a motion going through near the obstacles at fast speed. Moreover, the hazard regions are also represented as '0', because the only efficiency of the motion is considered (i.e. how fast the robot can reach the goal). In this case, the robot moves within the hazard regions at fast speed, and it might cause the some troubles. Therefore, it is necessary to consider not only the efficiency of the motion but also the safety of the motion in the motion planning.

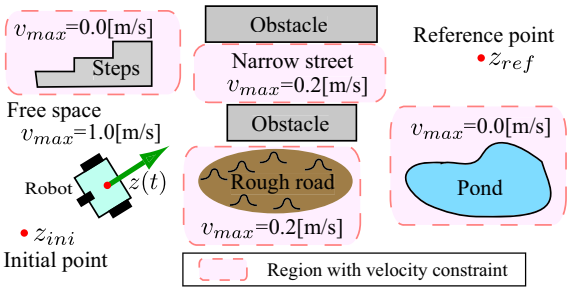

Fig. 1. Concept of regions with velocity constraint

The studies considering the safety of the autonomous locomotion are found in some literature. In paper [12], they proposed the Inevitable Collision State, representing the region where the collisions with the obstacles can not be avoided. The inevitable collision states are defined around each obstacle, and treated as the virtual obstacle in the motion planning to improve the safety of the motion. In other words, the obstacles are enlarged virtually to reduce the possibility of the collisions. In paper [13], an adaptive robot speed control method, considering the trade-off between the safety and the efficiency in navigation, is proposed. This method determines the admissible speed according to the uncertainties of the observation of the environment and the motion in the unknown local environment. Therefore, this work is different from the global motion planning as addressed in this study.

In this study, we propose a new motion planning method taking into account the regions with the velocity constraints, based on Global Dynamic Window Approach (GDWA)[8], [9] which is one of the standard motion planning methods. In the standard GDWA, a number of trajectories are predicted over the constant time based on the dynamic and kinematic constraints of the robot, and the optimal trajectory is selected from the predicted trajectories based on the cost function. Since the cost function consists Navigation Function (NF)[5], a local minima-free potential function, this method can generate the local minima-free trajectory. However the standard GDWA cannot take into account both the efficiency and the safety of the motion at the same time, as mentioned

K. Goto is with the Department of Mechanical Engineering and Intelligent Systems, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo, Japan goto@hi.mce.uec.ac.jp

K. Kon and F. Matsuno are with the Department of Mechanical Engineering and Science, Kyoto University, Yoshidahonmachi, Sakyoku, Kyoto, Kyoto, Japan k.kon@fs4.ecs.kyoto-u.ac.jp matsuno@me.kyoto-u.ac.jp

above. Therefore the proposed method modifies the standard GDWA so as to generate the efficient motion while ensuring the safety by taking into account the RVCs. Furthermore, we investigate the effectiveness of the proposed method by numerical examples and experiments.

II. Single robot navigation in the regions with velocity constraint

We propose the concept of the RVC, the regions imposed the admissible velocity of the robot. In other words, the robot has to satisfy the velocity constraint of the RVC while moving within the RVC. The velocity constraint in each RVC is registered by the operator in advance, based on the priori map information. For example, the hazard areas where the robot is not allowed to enter (e.g. steps, neighborhood of the pond) are registered as the regions with velocity constraint $v_{max} = 0$, and the area where the robot should reduce the velocity (e.g. rough road, narrow street) are registered as the regions with the velocity constraint $v_{max} < 1.0$ (see Fig. 1). Since the robot plans the motion according to the registered information, it is expected that the predicted hazard can be avoided (e.g. reduction of the shock if the robot collides an obstacle, passing smoothly at the complicated narrow road).

In this paper, we address a control problem of navigating the robot from the initial position z_{ini} to the reference position z_{ref} in the environment including the RVCs.

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \omega, \tag{1}$$

where v, ω are the translational and rotational velocity of the robot respectively, and $z := (x, y)$ and θ denote the measurable coordinate with respect to the global frame. The maximum value of translational velocity, rotational velocity, translational acceleration and rotational acceleration of the robot are defined by v_{max}, ω_{max}, $a_{v_{max}}$ and $a_{\omega_{max}}$ respectively. We assume that N_R RVCs exist in the environment, and define the j th RVC as $\mathbf{R}vc_j (j = 1, \ldots, N_R)$. Let $v_{j_{max}}$ be the admissible velocity of the j th RVC such that $v_{j_{max}} \leq v_{max}$. Hence, we consider the constraints only on the translation velocity, not on the rotational velocity of the robot, for simplicity. Specifically, we call the k th RVC, where $v_{k_{max}} = 0$ is imposed, as the k th obstacle region, and define as $\mathbf{O}_k (k = 1, 2, \cdots, N_O \leq N_R)$. Note that, the robot has to decelerate enough in the outside of the RVCs so as to satisfy the velocity constraint in the RVCs, because the maximum acceleration of the robot is considered.

To summarize the control objective in this study is to achieve equation (2) while satisfying the constraints in (3).

$$\lim_{t\to\infty} z(t) = z_{ref}, \tag{2}$$

$$\text{where, } z(t) \in \mathbb{R}^2 \setminus \mathbf{O}_k \ (k = 1, 2, \cdots, N_O)$$

$$\|\dot{z}(t)\| \leq v_{j_{max}}, \tag{3}$$

$$\text{if } z(t) \in \mathbf{R}vc_j \ (j = 1, 2, \cdots, N_R)$$

In the following sections, we propose a new motion planning method which achieves the control objective. The proposed motion planning method is divided into two parts : the first part is path planning method presented in Section III; and the second part is trajectory generation method presented in Section IV.

III. Path planning

In this section, we propose a new path planning method taking into account the RVCs based on NF, which has been also used in GDWA.

A. Path planning based on navigation function

NF is a grid-based local minima-free potential function computed by using wave-propagation technique starting at the reference position.

The outline to generate the NF is as follows. First, construct the grid map, representing the free-space as "0" and the obstacle as "1". We define a cell as (i, j) based on its index of row and column, and define the goal cell including the reference position as (i_g, j_g). Next, calculate the distance between the centroid of the cell $z_{ij} := (x_i, y_j)$ and the centroid of the goal cell $z_{i_g j_g} := (x_{i_g}, y_{j_g})$. Its distance is stored in each cell (i, j) as the information. In general, the distance between the cells is calculated by Manhattan Distance M_{ij} as follows:

$$\mathrm{M}_{ij} := \begin{cases} \infty & \text{if } (i, j) \text{ is an obstacle cell} \\ \mathrm{M}_{i'j'} + \|z_{ij} - z_{i'j'}\|_\infty & \text{otherwise,} \end{cases} \tag{4}$$

where a cell (i', j') denotes one of the 4-neighborhood of cell (i, j) (i.e. one of $(i-1, j)$, $(i+1, j)$, $(i, j-1)$ and $(i, j+1)$, storing the minimum value). Note that, if the cell (i, j) is an obstacle cell, the value much larger than the possible value of M_{ij} is stored.

The map with the distance in (4) is called as a NF (e.g. Fig. 3(a)). Since the NF is a local minima-free potential function, the shortest path to the reference position can be obtained by moving toward the cell having the smaller value.

B. Path planning considering the regions with velocity constraint

In the standard NF, the RVCs cannot be considered, because Manhattan Distance M_{ij} in (4) is used to evaluate the path. In other words, the velocity constraint in each region is not reflected to the path planning, because only the distances between cells are used to evaluate the path. It is reasonable to use the distance based potential in the environment except the RVCs, because the robot can move at maximum velocity in all free-space cells. However, in the environment including the RVCs, it is not reasonable to evaluate the path by the distance based potential in (4), because the admissible velocity may be different at each region. In order to take into account the velocity constraint of each RVC in the path planning, we modify the NF so that the stored value in each cell represents the time required to move (i.e. the information based on the admissible velocity). More precisely, we change the criteria M_{ij} for evaluating paths to N_{ij}, a time required to move from cell (i, j) to cell (i_g, j_g).

The outline to generate the proposed NF is as follows. First, store the information of the velocity constraint of the corresponding RVC to each cell. We define the stored velocity constraint in cell (i, j) as $v_{ij_{max}}$, and in the cell (i', j') as $v_{i'j'_{max}}$. As shown in Fig. 2(a), we define the time

required to move from z_{ij} to $z_{i'j'}$, in the case of both the cell (i,j) and (i',j') are the free-space cell, as the unit time (i.e. $(\|z_{ij}-z_{i'j'}\|/v_{max}:=1)$. The value $N_{ij}=N_{i'j'}+1$ is stored in the cell (i,j). In the case of two cells having same velocity constraints $v_{k_{max}}=0.8v_{max}$ as shown in Fig. 2(b), the time required to move from z_{ij} to $z_{i'j'}$ can be computed by $\frac{v_{max}}{v_{k_{max}}}=1.25$. Then the value $N_{ij}=N_{i'j'}+1.25$ is stored in the cell (i,j). Next we consider the case where the constrained velocity $v_{ij_{max}}$ of cell (i,j) and the constrained velocity $v_{i'j'_{max}}$ of cell $(i,'j')$ differ from each other, as shown in Fig. 2(c) ($v_{i'j'_{max}}=v_{max}$ and $v_{ij_{max}}=0.5v_{max}$). In this case, since the movement distance in each cell becomes half, we can compute the time by $\frac{\|z_{ij}-z_{i'j'}\|/2}{v_{ij_{max}}}+\frac{\|z_{ij}-z_{i'j'}\|/2}{v_{i'j'_{max}}}=\frac{1}{2}\frac{v_{max}}{v_{ij_{max}}}+\frac{1}{2}\frac{v_{max}}{v_{i'j'_{max}}}=\frac{1}{2}+\frac{1}{2}2=1.5$. Then $N_{ij}=N_{i'j'}+1.5$ is stored in the cell (i,j).

To summarize the time N_{ij} required to move M_{ij} between cell (i,j) and cell (i_g,j_g) can be written as follows:

$$N_{ij}:=\begin{cases}\infty & \text{if } (i,j) \text{ is an obstacle cell}\\ N_{i'j'}+\frac{1}{2}\frac{v_{max}}{v_{ij_{max}}}+\frac{1}{2}\frac{v_{max}}{v_{i'j'_{max}}} & \text{otherwise.}\end{cases} \quad (5)$$

Fig. 3 shows an example of the path planning. Fig. 3(a) shows an example of the standard NF in the environment without the RVC, and Fig. 3(b) shows an example of proposed method in an environment including Region 1 with constrained velocity $v_{1_{max}}=0.25v_{max}$ and Region 2 with constrained velocity $v_{2_{max}}=0.5v_{max}$. The value in each cell shows the stored value (i.e. M_{ij} in (a) and N_{ij} in (b)). The arrow in each cell shows the gradient of the NF, and the dashed line shows the shortest path obtained by the NF. Compared with the path planned by the standard method and the proposed method, we can see that the path in Fig. 3(b) goes through the Region 2, although the path in Fig. 3(a) goes through the Region 1. Therefore, the RVCs are taking into account in the path planning correctly.

The proposed method can plan a path converging to the reference position without getting stuck in local minima, since the procedure is same as the standard one.

IV. Trajectory Generation

In this section, we propose a new trajectory generation method taking into account the RVCs. The proposed method is based on the GDWA, which is one of the standard trajectory generation methods.

A. Standard Global Dynamic Window Approach

The GDWA is an algorithm extending Dynamic Window Approach [11], an obstacle avoidance method, to a global trajectory generation method. The GDWA uses the NF in (4) to avoid the local minima.

The outline of the GDWA can be summarized as follows. First, make p tuples (a_v,a_ω) of translational acceleration $|a_v|\le a_{v_{max}}$ and rotational acceleration $|a_\omega|\le a_{\omega_{max}}$ discretely. Note that, we do not make a tuple whose predicted velocities $v(t+T)$ or $\omega(t+T)$ violates the constraints $v\in[0,v_{max}]$ or $\omega\in[-\omega_{max},\omega_{max}]$, where, T[s] is the time length evaluating predicted trajectories. Then, select one of the

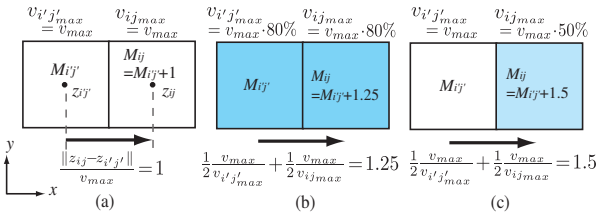

Fig. 2. Examples of the calculation of N_{ij} in (5)

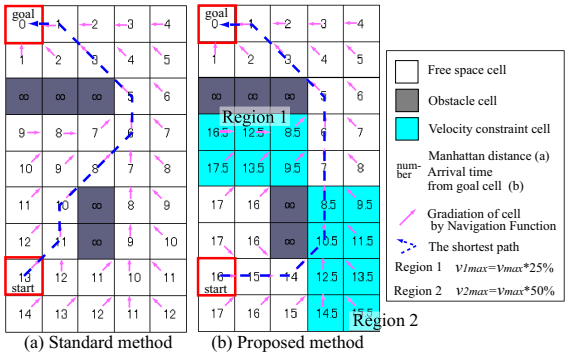

Fig. 3. Example of path planning

tuples, and predict the translational velocity $v(t+\Delta t)$ and rotational velocity $\omega(t+\Delta t)$ at $t+\Delta t$[s] ($\Delta t\le T$) based on the selected tuple (a_v,a_ω). And the future position $z(t+\Delta t)$ and bearing $\theta(t+\Delta t)$ of the robot are predicted based on the kinematics in (1). By iterating the above procedure for every tuples, the GDWA predicts the trajectories over T[s]. The predicted trajectories are evaluated based on the cost function in (6). Finally, one of the trajectories maximizing the Ω is selected as an optimal one.

$$\Omega = \alpha NF1+\beta vel+\gamma goal+\delta\Delta NF1, \quad (6)$$

where $NF1$ is a function increased if the predicted velocity is aligned with the gradient of the NF in (4) at the predicted position. *vel* is a function increasing as the predicted translational velocity, *goal* is a function increased if the predicted bearing is aligned with the direction to the reference position at the predicted position. $\Delta NF1$ is a function indicates how much a motion is expected to reduce the value of $NF1$ during next time step. α, β, γ and δ are the parameters to modify the behavior of the robot.

Note that, since the trajectory of only short time interval T[s] is generated, the GDWA obtains a trajectory from the initial point to the goal point by iterating the above procedure until converging the reference position.

B. Trajectory generation considering the regions with velocity constraint

In this section, we propose a new trajectory generation method taking into account the RVCs, by modifying the standard GDWA described in the previous section.

In order to generate the trajectory in the environment including the RVCs, it is necessary to consider the velocity constraints of each RVC. Moreover, as mentioned in Section II, the robot has to reduce the velocity enough in advance to satisfy the velocity constraints of each RVC. We call the region where the robot is required to reduce the velocity, as *The Ac/Deceleration region* (AD region). Therefore the

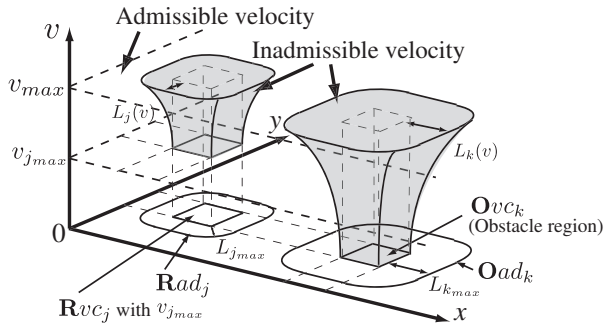

Fig. 4. Conceptual diagram of regions with velocity constraint

```
1: while z(t) does not reach to z_ref do
2:   for n {n : total number of candidate of suboptimal trajectories} do
3:     for p {p : the number of tuples (a_v, a_ω)} do
4:       for τ = Δt to T step Δt do
5:         compute v(t+τ), ω(t+τ) by tuple (a_v, a_ω)
6:         compute z(t+τ) based on kinematics in (1)
7:         if z(t+τ) ∈ Rad_j or Rvc_j then
8:           compute ℓ_j, V(ℓ_j), and v_{j_max} in (7), (8)
9:           if v(t+τ) > min[V(ℓ_j), v_{j_max}] then
10:            recompute a_{v_new} in (9)
11:            recompute v(t+τ) and z(t+τ) by a_{v_new}
12:          end if
13:        end if
14:      end for
15:      compute Ω in (6)
16:    end for
17:  end for
18:  choose n candidates of suboptimal trajectories, and update t as
     t = t+T
19: end while
```

Fig. 5. The pseudo-code of the proposed trajectory generation method

trajectory generation method has to take into account not only the RVCs but also the AD regions.

To make this more precise, we define the AD regions and the constrained velocities in the AD regions. Let the robot moves toward the jth RVC, where the velocity constraint $v_{j_{max}}$ is imposed, at $v \leq v_{max}$. In this case, the minimum distance required to reduce the translational velocity enough can be described as follows:

$$L_j(v) := \frac{1}{2a_{v_{max}}}(v^2 - v_{j_{max}}^2). \tag{7}$$

Specifically, $L_{j_{max}} := L_j(v_{max})$ denote the minimum distance at $v = v_{max}$, and call the region surrounded by $L_{j_{max}}$ as the j th AD region ($\mathbf{R}ad_j$). If the robot is within the j th AD region, the robot is imposed the following velocity constraint according to the distance ℓ_j between the robot and the j th AD region.

$$V(\ell_j) = \sqrt{v_{j_{max}}^2 + 2\ell_j a_{v_{max}}}. \tag{8}$$

Fig. 4 illustrates the admissible velocity space in the environment including the RVCs and the obstacles. The left object in this figure illustrates the j th RVC ($\mathbf{R}vc_j$), and the right one illustrates the k th obstacle ($\mathbf{O}ad_k$). The shaded areas show the AD regions defined in (7), and indicate that the constrained velocity will be violated, even though the robot decelerates with the maximum deceleration. Since the obstacles are also represented as the RVCs with the constraints 0[m/s], the AD regions of the obstacles become much larger than the ones of the RVCs ($v_{j_{max}} > 0$). The

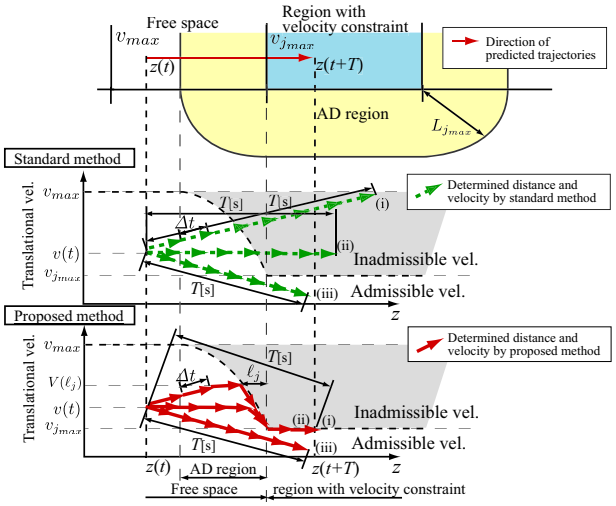

Fig. 6. Prediction of trajectories around the RVC and its AD region

regions surrounded by $L_{j_{max}}$ on the xy plane shows the j th AD region ($\mathbf{R}vc_j$) and the k th obstacle ($\mathbf{O}ad_k$). If the robot is present in these regions, the robot can satisfy the velocity constraints of each RVC by decelerating with the acceleration as follows:

$$a_{v_{new}} = \frac{V(\ell_j) - v(t)}{\Delta t}, \tag{9}$$

where ℓ_j is the distance between the robot and the RVC.

The proposed trajectory generation method taking into account the RVCs and AD regions is described in Fig. 5. From line 3 to line 6, the velocities $v(t+\tau)$, $\omega(t+\tau)$ and the position $z(t+\tau), (\Delta t \leq \tau \leq T[\mathrm{s}])$ are predicted based on selected tuple (a_v, a_ω). In line 7, judge whether the predicted trajectory belongs to an AD region or a RVC. If the predicted trajectory belongs to an AD region or a RVC, then compute the constrained velocity in (8). If the predicted velocity violates the constrained velocity in (8), then the acceleration in (9) is calculated and the trajectory is recalculated based on the acceleration in (9) (From line 9 to line 13). By iterating above procedure for all tuples, the trajectories satisfying the velocity constraints in each RVC are predicted. The predicted trajectories are evaluated based on the cost function in (6), and n trajectories are selected in descending order of the cost, as the candidates of the optimal one (From line 14 to line 18). Until at least one predicted trajectory converges to the reference position, the above procedure is iterating (line 2). Note that, the $NF1$ and $\Delta NF1$ in (6) are calculated based on the result in Section III-B.

Remark 1: In general, in order to obtain the optimal trajectory, we have to search the all of possible patterns. However, it requires the heavy computational burden, since the computational burden increases as the combination of the number of tuples (a_v, a_ω). One way to improve the computational efficiency is to use the local optimal trajectory obtained at each iteration, as in the standard GDWA. However, the optimality of the global trajectory cannot be considered in this method. Therefore, we modify the algorithm so that the trade-off between the computational burden and the optimality of the trajectory can be changed depending on the problem (i.e. the number of n).

TABLE I

PARAMETER SETTING

Notation	Value [unit]
Robot's width	0.35 [m]
Max translational vel. :v_{max}	0.3 [m/s]
Max rotational vel. :ω_{max}	0.6 [rad/s]
Max translational acc. :$a_{v_{max}}$	0.05 [m/s^2]
Max rotational acc. :$a_{\omega_{max}}$	0.4 [rad/s^2]
Grid size	0.5[m] × 0.5[m]
Width and height of map	2.5[m] × 4.5[m]

Fig. 6 illustrates an example of replanning the predicted trajectory in the AD region (i.e. from line 9 to line 13 in Fig. 5). Let the robot goes straight to the j th RVC where the velocity constraint $v_{j_{max}}$ is imposed. Let further the translational accelerations are discretized into three patterns ((i)$a_v > 0$, (ii)$a_v = 0$ and (iii)$a_v < 0$). The dashed arrows show the trajectory predicted by the standard GDWA, and the solid arrows show the trajectory predicted by the proposed method. In the standard GDWA, it can be seen that the velocity constraint is violated in the case of (i) and (ii). On the other hands, in the proposed method, the trajectories satisfy the velocity constraint of the RVC, by calculating the acceleration in (9) and recalculating the trajectory.

V. NUMERICAL EXAMPLE

In this section, we investigate the effectiveness of the proposed method by numerical examples.

The specification of the robot and the environment settings used in simulation is shown in Table I and Fig. 3. We show the simulation results about three kinds of environments (Environment 1, 2 and 3). The environment 1 has the velocity constraint $v_{1_{max}} = 0.1$[m/s] at Region 1 in the Fig. 3, the environment 2 has the velocity constraint $v_{1_{max}} = v_{2_{max}} = 0.15$[m/s] at Region 1 and Region 2 respectively, and the environment 3 has the velocity constraint $v_{1_{max}} = 0.075$[m/s] at Region 1 and the velocity constraint $v_{2_{max}} = 0.15$[m/s] at Region 2. In these simulations, the translational acceleration a_v is discretized into five patterns $[-a_{v_{max}}, -a_{v_{max}}/2, 0, a_{v_{max}}/2, a_{v_{max}}]$, and the rotational acceleration a_ω is also discretized into five patterns $[-a_{\omega_{max}}, -a_{\omega_{max}}/2, 0, a_{\omega_{max}}/2, a_{\omega_{max}}]$. Other parameters are $\alpha = 0.3$, $\beta = 0.2$, $\gamma = 0.05$, $\delta = 0.45$, $T = 2$[s], $\Delta t = 0.2$[s] and $n = 20$. Note that the simulations are carried out using Matlab R2009b on a PC (CPU:Core2Duo 2.80GHz and RAM:4.00GB).

Fig. 7-Fig. 10 show the simulation results. Fig. 7 and Fig. 8 show the simulation results in the environment 1 based on the standard method and the proposed one respectively. Fig. 9 and Fig. 10 show the simulation results in the environment 2 and 3 based on proposed method. The figure (a) in each figure shows the trajectory of the robot, the solid lines show the trajectories predicted based on the acceleration tuples, and the dashed line shows the optimal trajectory. The value in each cell shows the M_{ij} in (4) (shown in Fig. 7(a)) or the N_{ij} in (5) (shown in Fig. 8(a)-Fig. 10(a)). Moreover, the figure (b) in each figure shows the time response of translational velocity. Note that, the dots P_{k1}, P_{k2} and P_{k3} (k: the number of simulations) are plotted on the optimal trajectories to show

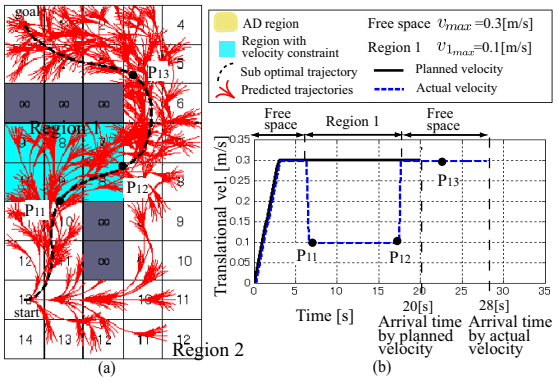

Fig. 7. Simulation result of standard method (Environment 1)

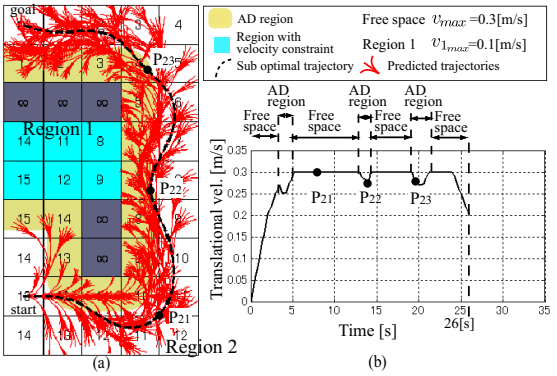

Fig. 8. Simulation result of proposed method (Environment 1)

the relation between the figure (a) and (b). From Fig. 8-Fig. 10, we can see that the obtained trajectory converges to the reference position even in the environment including the RVCs. Furthermore, we can also see that the robot reduces the velocity enough at the AD regions to satisfy the velocity constraint of each RVC.

In the standard method, the robot goes through the Region 1 at high speed as shown in Fig. 7. This is because the standard method does not consider the velocity constraint of the Region 1. However, if the robot follows this trajectory, the velocity of the robot has to be reduced in the Region 1 for safety, as shown in Fig. 7(b). In this case, it might take at least 28[s] to reach the reference position. In contrast, as shown in Fig. 8, the proposed method generates the trajectory which goes through the Region 2, where is a free-space, and it takes about 26[s]. Therefore the proposed method can generate the more efficient trajectory even in the environment including RVCs.

Moreover, the standard GDWA generates a trajectory going through near the obstacles at high speed as shown in Fig. 7 (at P_{11}, P_{13}). On the other hand, in the proposed method, the AD regions are present around not only the RVCs but also the obstacles. Therefore, the robot tends to keep more distance from the obstacles, so as not to reduce the velocity extremely (i.e. P_{21} in Fig. 8, P_{33} in Fig. 9). Even if the robot passes through near the obstacles, the robot reduces the velocity enough to ensure the collision avoidance (i.e. P_{23} in Fig. 8, P_{43} in Fig. 10). As described above, it is said that the proposed method can generate the trajectory taking into account the efficiency while ensuring its safety.

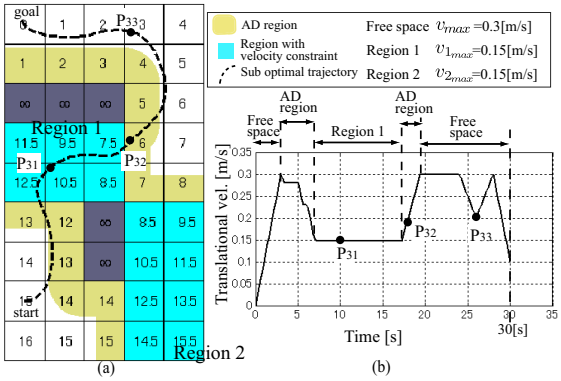

Fig. 9. Simulation result of proposed method (Environment 2)

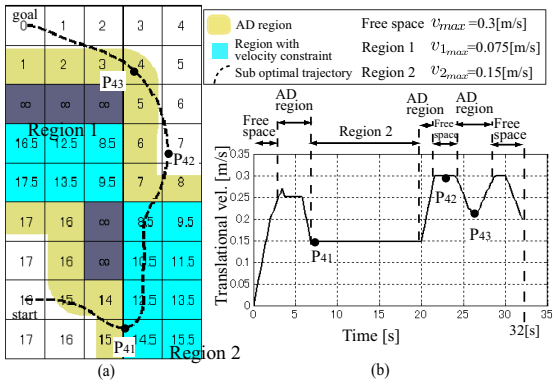

Fig. 10. Simulation result of proposed method (Environment 3)

VI. Experiments

The proposed motion planning method is applied to a mobile robot platform "beego" (TechnoCraft), which is a two wheeled skid-steer robot with one caster wheel. The specification of the robot and the environmental settings are shown in Table I, same as the simulation. In these experiments, the trajectory is generated in advance, and the robot follows the planned trajectory. Note that the coordinate of the robot is measured by dead-reckoning, and the algorithm is implement by using RT-Middleware[14].

We show one of our experimental results, which is carried out in the environment 3 (see Fig. 11). The left figure shows the trajectories of vehicle, the dashed line shows the reference trajectory generated in advance, and the solid line shows the robot one. The right figure shows the time response of translational velocity. From this figure, we can see that the robot converges to the reference position while satisfying the constraints on the translational velocity in each RVC. Moreover, it can be seen that the robot reduces the translational velocity near the obstacles from the time response of the translational velocity (i.e. P_{53} in Fig. 11).

VII. Conclusion

In this paper, we proposed the concept named "The Region with Velocity Constraint" which intends to improve the safety of the autonomous locomotion in the real environment. We also proposed the new path planning and trajectory generation method enables to navigate the robot even in the environment including the RVCs. The proposed method generates the trajectory navigating the robot to the reference

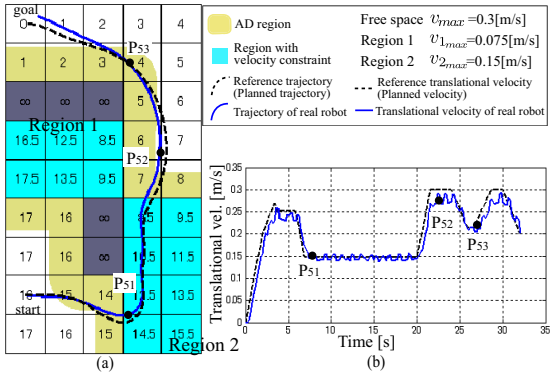

Fig. 11. The example of experimental result (Environment 3)

position while satisfying the constraints of each RVC. One of the features of the proposed method is that, it can generate the safe trajectory, because the motion with high velocity near the obstacles and the RVCs is not allowed, Furthermore, the effectiveness of the proposed method has been investigated by the numerical examples and experiments.

VIII. Acknowledgments

The authors would like to thank Prof. Yasuyoshi Yokokohji, Kobe University, for his useful comments on motion planning method. This work was supported by commissioned business of New Energy and Industrial Technology Development Organization (NEDO).

References

[1] B. Graf, M. Hans and R. D. Schraft : Mobile robot assistant; *IEEE Journal of Robotics and Automation*, 11(2), pp. 67-77 (2004)

[2] H. Iwata and S. Sugano : Design of human symbiotic robot TWENDY-ONE; *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 580-586 (2009)

[3] S. Wirth and J. Pellenz : Exploration Transform: A stable exploring algorithm for robots in rescue environments; *Proc. of IEEE int. workshop on Safety, Security, and Rescue Robotics*, 28-P3 (2007)

[4] M. Ohkita *et al.* : Traveling control of the autonomous mobile wheelchair DREAM-3 considering correction of the initial position; *Proc. of the 2004 47th IEEE Int. Midwest Symposium on Circuits and Systems*, 3, pp. 215-218 (2004)

[5] J. Latombe : Robot motion planning; *Kluwer Academic Publishers* (1991)

[6] S. Ge and F. Lewis : Autonomous mobile robots; *Taylor & Francis Group* (2006)

[7] S. M. LaValle : Rapidly-exploring random trees: A new tool for path planning; Computer Science Dept., Iowa State Univ., TR 98-11 (1998)

[8] O. Brock and O. Khatib : High-speed navigation using the global dynamic window approach; *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1, pp. 341-346 (1999)

[9] P. Ögren and N. Leonard : A convergent dynamic window approach to obstacle avoidance; *IEEE Trans. on Robotics*, 21(2), pp. 188-195 (2005)

[10] J. Borenstein and Y. Koren : The vector field histogram fast obstacle avoidance for mobile robots; *IEEE Trans. on Robotics and Automation*, 7(3), pp. 278-288 (1991)

[11] D. Fox, W. Burgard and S. Thrun : The dynamic window approach to collision avoidance; *IEEE J. Robotics and Automation Magazine*, 4(1), pp. 23-33 (1997)

[12] T. Fraichard and H. Asama : Inevitable collision states, A step towards safer robots?; *J. of Advanced Robotics*, 18(10), pp. 1001-1024 (2004)

[13] J. Miura, Y. Negishi, Y. Shirai : Adaptive robot speed control by considering map and motion uncertainties; *J. of Robotics and Autonomous Systems*, 24, pp. 110-117 (2006)

[14] N. Ando *et al.* : RT-Middleware: Distributed Component Middleware for RT (Robot Technology); *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 3555-3560 (2005)