# Real-Time Optimization for Mobile Robot Based on Algorithmic Control

Tomoaki Kobayashi*, Junichi Maenishi*, Joe Imae* and Guisheng Zhai*

*Department of Mechanical Engineering, Osaka Prefecture University, Sakai, Osaka 599-8531, Japan
(Tel: +81-72-254-9218; Fax: +81-72-254-9904; Email:kobayasi@me.osakafu-u.ac.jp)

**Abstract:** In this paper, a real-time optimization method for nonlinear dynamical systems is proposed. The proposed method is based on the algorithms of numerical solutions for optimal control problems. We deal with a real-time collision-free motion control of a nonholonomic mobile robot, which has input restrictions of actuators. The effectiveness of the algorithmic method is demonstrated through numerical and experimental results. The mobile robot which we have developed is able to avoid moving obstacles skillfully. Therefore the proposed controller works well in real time.

**Keywords:** Real time optimization, mobile robot, nonlinear system, algorithmic design.

## 1. INTRODUCTION

In this paper, a real-time optimization method for nonlinear dynamical systems is proposed. We deal with a real-time collision-free motion control of a nonholonomic mobile robot that is practically important and theoretically difficult to solve. The proposed method is based on the algorithms of numerical solutions of optimal control problems.

When solving an optimal control problem for nonlinear systems, many iterative calculations are required[1],[2]. Because much computation time is required, the conventional controller is designed in a off-line fashion. It is true that the computation time has been shortened by development of the recent computer technology. However, it is not yet sufficient in order to solve the nonlinear optimum control problem in real time for mechanical systems, which requires fast sampling.

In this paper, to treat input restrictions of the actuators, we advanced our conventional real-time argorithmic control method[14]. This controller is not designed directly, but it mounts the algorithm as a controller. This is called ‘ Algorithmic Controller ’. In the proposed algorithmic method, the evaluation section of performance index is moved on time. Therefore, our method is similar to the approach of Model Predictive Control (MPC)[4]. The research of the nonlinear model predictive control (NMPC)[5],[6] has lately attracted considerable attention. Some efficient algorithm was proposed for real time control[7],[11]. However, previous method has a problem on its robustness because of using the open-loop solution. On the other hand, our method is based on Riccati-equation-based (REB) algorithm for nonlinear optimal control problems[3]. Therefore, online feedback control is possible, because the REB algorithm can calculate feedback gain based on Riccati equation.

The effectiveness of proposed algorithmic method is demonstrated through numerical and experimental results. It is applied to a differential-driven mobile robot, where there are input restrictions of actuators and there exist obstacles. The mobile robot which we developed is able to avoid moving obstacles skillfully. The numerical and experimental results show the proposed controller works well in real time, even in the case of including moving obstacles.

The outline of the paper is as follows. In Section 2, the nonlinear optimal control problems are formulated. Also, one of the computational methods for optimal control problems is given with its convergence property. In Section 3, based on the computational method, our algorithmic design method is proposed. In Sections 4 and 5, simulation and experiment results are given in order to demonstrate the effectiveness and practicability of our approach, where the collision-free motion control of a nonholonomic mobile robot is dealt with.

## 2. OPTIMAL CONTROL PROBLEM

### 2.1. Formulation

System equation, initial condition, and performance index are given as follows.

$$\dot{x}(t) = f(t, x(t), u(t)) \tag{1}$$

$$x(t_0) = x_0 \in \Re^n \tag{2}$$

$$J = G(x(t_1)) + \int_{t_0}^{t_1} L(t, x(t), u(t))dt \tag{3}$$

where $t_0$, $t_1$ are initial/terminal time given. Then, our goal is to find a controller minimizing the performance index $J$ over a time interval $[t_0, t_1]$. Here, denote the state variable by $x(t) = [x_1(t), \cdots, x_n(t)]^T \in \Re^n$, and the input variable by $u(t) = [u_1(t), \cdots, u_r(t)]^T \in R^r$. Based on the problem formulation (1) to (3), we describe our on-line computational design method, that is to say, algorithmic design method.

Before that, we give some preliminaries. Whether or not the algorithmic design method works would depends on how effective the algorithm is in searching the numerical solutions of optimal control problems. In this paper, we adopt one of the so-called Riccati-equation based algorithms (REB algorithms), which is known to be reliable and effective in searching numerical solutions. One of the characteristics is the use of feedback structure in the process of computation of solutions. Details are given later.

## 2.2. Riccati-equation based algorithm

Under the problem formulation (1) to (3), we describe an iterative algorithm for the numerical solutions of optimal control problems, based on Riccati differential equations. In this respect, the algorithm falls in the category of optimal control algorithms, such as the REB algorithms presented in [1], [3], [8], [9], [10], [12] and [13].

**Assumptions**

Let $x : [t_0, t_1] \to \Re^n$ be an absolutely continuous function, and $u : [t_0, t_1] \to \Re^r$ be an essentially bounded measurable function. For each positive integer $j$, let us denote by $AC^j$ all absolutely continuous functions: $[t_0, t_1] \to \Re^j$, and by $L^j_\infty$ all essentially bounded measurable functions: $[t_0, t_1] \to \Re^j$. Moreover, we define the following norms on $AC^j$ and $L^j_\infty$ respectively:

$$\begin{aligned} \| x \| &= \max \mid x(t) \mid \text{for } x \in AC^j, t \in [t_0\ t_1] \\ \| y \| &= \text{ess sup} \mid y(t) \mid \text{for } y \in L^j_\infty, t \in [t_0\ t_1] \end{aligned}$$

where the $\mid \bullet \mid$ is used to denote Euclidean norms for vectors.

Now, we make some assumptiions.

(i) $G : \Re^r \to \Re^1, f : \Re^1 \times \Re^n \times \Re^r \to \Re^n, L : \Re^1 \times \Re^n \times \Re^r \to \Re^1$ are continuous in all their arguments, and their partial derivatives $G_x(x), f_x(t,x,u), f_u(t,x,u), L_x(t,x,u)$ and $L_u(t,x,u)$ exist and are continuous in all their arguments.

(ii) For each compact set $U \subset \Re^r$ there exists some $M_1 \in (0, \infty)$ such that

$$\mid f(t,x,u) \mid \leq M_1(\mid x \mid +1) \tag{4}$$

for all $t \in \Re^1, x \in \Re^n,$ and $u \in U$.

**Algorithm**

*Step A-0:* Let $\beta \in (0,1)$ and $M_2 \in (0,1)$. Select arbitrarily an initial input $u^0 \in L^r_\infty$.

*Step A-1:* $i = 0$

*Step A-2:* Calculate $x^i(t)$ with $u^i(t)$ from the equation (1).

*Step A-3:* Select $A^i \in \Re^{n\times n}$, $B^i_{11} \in L^{n\times n}_\infty$, $B^i_{12} \in L^{n\times r}_\infty$ and $B^i_{22} \in L^{r\times r}_\infty$ so that Kalman's sufficient conditions for the boundedness of Riccati solutions hold, that is, for almost all $t \in [t_0, t_1]$,

$$\begin{gathered} A^i \geq 0, B^i_{22}(t) > 0, \\ B^i_{11}(t) - B^i_{12}(t){B^i_{22}}^{-1}(t){B^i_{12}}^T(t) \geq 0 \end{gathered} \tag{5}$$

where $A^i$, $B^i_{11}$ and $B^i_{22}$ are symmetric and $(\cdot)^T$ means the transpose of vectors and matrices. We solve $\delta x^i(t)$, $K^i(t)$, $r^i(t)$ from (7), (7), and (8) below,

$$\begin{aligned} \delta\dot{x}(t) = \{&f_x(t,x^i,u^i) + f_u(t,x^i,u^i){B^i_{22}}^{-1} \\ &\times(f^T_u(t,x^i,u^i)K(t) - {B^i_{12}}^T)\}\delta x(t) \\ &+f_u(t,x^i,u^i){B^i_{22}}^T(f^T_u(t,x^i,u^i)r(t) \\ &-L^T_u(t,x^i,u^i)),\ \delta x(t_0) = 0 \end{aligned} \tag{6}$$

$$\begin{aligned} \dot{K}(t) = &-K(t)f_x(t,x^i,u^i) - f^T_x(t,x^i,u^i)K(t) \\ &+B^i_{11} + (K(t)f_u(t,x^i,u^i) - B^i_{12}){B^i_{22}}^{-1} \\ &\times({B^i_{12}}^T - f^T_u(t,x^i,u^i)K(t)),\ K(t_1) = -A^i \end{aligned} \tag{7}$$

$$\begin{aligned} \dot{r}(t) = &-f^T_x(t,x^i,u^i)r(t) + L^T_x(t,x^i,u^i) + \{B^i_{12} \\ &-K(t)f_u(t,x^i,u^i)\}{B^i_{22}}^{-1}(-L^T_u(t,x^i,u^i)) \\ &+f^T_u(t,x^i,u^i)r(t)),\ r(t_1) = -G(x(t_1)) \end{aligned} \tag{8}$$

and determine $\delta u^i$ from the following.

$$\begin{aligned} \delta u^i(t) = {B^i_{22}}^{-1}\{&((f^T_u(t,x^i,u^i)K^i(t) - {B^i_{22}}^T))\delta x^i \\ &+f^T_u(t,x^i,u^i)r^i(t) - L^T_u(t,x^i,u^i)\} \end{aligned} \tag{9}$$

*Step A-4:* Determine $(\tilde{x}^i, \tilde{u}^i)$ satisfying

$\dot{x}(t) = f(t,x(t),u(t)), x(t_0) = x_0 \in \Re^n$

$H^i(t,(x-x^i),(u-u^i),p^i)$

$\qquad = \max_{v\in\Re^r} H^i(t,(x-x^i),(v-v^i),p^i)$

where

$$\begin{aligned} &H^i(t,\delta x,\delta u,p) \\ &\quad= -\{L_x(t,x^i,u^i)\delta x + L_u(t,x^i,u^i)\delta u \\ &\quad+\tfrac{1}{2}(\delta x^T B^i_{11}\delta x + 2\delta x^T B^i_{12}\delta u + \delta u^T B^i_{22}\delta u)\} \\ &\quad+p^T(f_x(t,x^i,u^i)\delta x + f_u(t,x^i,u^i)\delta u) \end{aligned}$$

and $p^i$ is a solution of the followings.

$\dot{p}(t) = -f^T_x(t,x^i,u^i)p(t) + L^T_x(t,x^i,u^i)$

$p(t_1) = -G^T_x(x(t_1))$

*Step A-5:* $\alpha_i = 1$.

*Step A-6:* Set $u^{i+1}(t) = u^i(t) + \alpha_i\delta u^i(t) + \alpha_i^2(\tilde{u}^i(t) - u^i(t) - \delta u^i(t))$. if (10) holds, go to Step 7. Otherwise, set $\alpha_i = \beta\alpha_i$ and repeat Step 6.

$$\begin{aligned} &J(u^{i+1}) - J(u^i) \leq \alpha_i M_2\{G(x_i(t_1))\delta x(t_1) \\ &+\int_{t_0}^{t_1}(L_x(t,x^i,u^i)\delta x^i + L_u(t,x^i,u^i)\delta u^i)\} \end{aligned} \tag{10}$$

*Step A-7:* Set $i = i+1$, and go to Step A-3. Repeat Step A-3 to Step A-7 until the performance index $J$ converges. Here, the integer $i$ represents the number of iterations.

## 2.3. Convergence

We can prove the convergence property of the algorithm described in the previous subsection. The theorem below tells us that accumulation points generated by Algorithm, if they exist, satisfy the necessary conditions for optimality. See [3] for more details.

**Theorem (convergence)**

Let $\{A^i\}^\infty_{i=0}$, $\{B^i_{11}\}^\infty_{i=0}, \{B^i_{12}\}^\infty_{i=0}, \{B^i_{22}\}^\infty_{i=0}, \{u^i\}^\infty_{i=0}$, $\{\tilde{u}^i\}^\infty_{i=0}$ and $\{\delta u^i\}^\infty_{i=0}$ be sequences generated by the above-mentioned algorithm. Suppose that there exists $M_4 \in (0,\ \infty)$ such that, for almost all $t \in [t_0,\ t_1]$,

$|\tilde{u}^i| \le M_4$ and $|\delta u^i| \le M_4$, $i = 0,1,2,\cdots$ and also suppose that exist $\bar{A} \in \Re^{n\times n}$, $\bar{B}_{11} \in L_\infty^{n\times n}$, $\bar{B}_{12} \in L_\infty^{n\times r}$, $\bar{B}_{22} \in L_\infty^{r\times r}$, $\bar{u} \in L_\infty^r$ and a sequence $N \subset (0,1,2,3,\cdots)$ such that $\bar{A} \ge 0$, $\bar{B}_{11}(t) - \bar{B}_{12}(t)(\bar{B}_{22}(t))^{-1}(\bar{B}_{12}^T(t)) \ge 0$ for almost all $t \in [t_0, t_1]$

$\lim_{i\in N} A^i = \bar{A}$ in the norm of $\Re^{n\times n}$

$\lim_{i\in N} B_{11}^i(t)(\text{resp.}, B_{12}^i(t), B_{22}^i(t))$

$= \bar{B}_{11}(t)(\text{resp.}, \bar{B}_{12}(t), \bar{B}_{22}(t))$

in the norm of $L^{n\times n}(\text{resp.}, L_\infty^{n\times r}, L_\infty^{r\times r})$

$\lim_{i\in N} u^i(t) = \bar{u}(t)$ in the norm of $L_\infty^r$

Here, $\bar{A}$, $\bar{B}_{11}$ and $\bar{B}_{22}$ are symmetric. Then, $\bar{u}(t)$ satisfies the weak necessary condition for optimality

$H_u(t, \bar{x}(t), \bar{u}(t), \bar{p}(t)) = 0$ a.e.in $t \in [t_0, t_1]$

where $H(t,x,u,p) = -L(t,x,u) + p^T f(t,x,u)$, $\bar{x}(t)$ is a solution of (1) with $\bar{u}$, and $\bar{p}(t)$ is a solution of the followings.

$$\begin{aligned} \dot{p}(t) &= -f_x^T(t,\bar{x},\bar{u})p(t) + L_x^T(t,x,u) \\ p(t_1) &= -G_x^T(\bar{x}(t_1)) \end{aligned}$$

## 3. ALGORITHMIC DESIGN

The key idea is very simple. Roughly speaking, all we have to do is to proceed with such above-mentioned algorithm by one iteration only, periodically. This means that as time passes the number of iterations increases. That is, through sufficiently large number of iterations, it could be expected to eventually reach the possible optimal solutions[14]. More detailed explanations are given in the following description.

From a practical point of view, the calculation time for one-iteration-ahead solution is assumed to be equal to $\Delta T$[ms] (or less than $\Delta T$). The calculation time $\Delta T$ plays a key role in our design method. We here describe how well the algorithmic controller works, together with Fig. 1.

**Algorithm**

Step B-1: Measure an actual state $x_0$, and select arbitrarily an initial input $u^0$. Set the unit of calculation time $\Delta T$[ms], and apply the input $u^0$ to the plant over the interval of the first unit time of calculation. During the interval (say, Section 1), proceed with two kinds of calculation. One is to predict the one-unit-time-ahead state $\hat{x}_1$ through system equation (1) with the initial state $x_0$, and the other is to calculate over $[\Delta T, t_1]$ the one-iteration-ahead solution (say, Optimal trajectory 1) with the updated initial state $\hat{x}_1$ as a new initial state. Then, denote by $k^1$ the feedback gain

$$-B_{22}^{-1}\{f_u(t,x,u)^T K + B_{12}^T\},$$

and by $u^1$ the input associated with Optimal trajectory 1.

Step B-2: Measure the actual state $x_1$, and apply the input $u^1$ together with the feedback gain $k^1$ to the plant over the interval of the second unit time of calculation. During such interval (say, Section 2), proceed with two kinds of computation. One is to predict the one-unit-time-ahead state $\hat{x}_2$ through the system equation (1) with the state $x_1$. The other is to calculate over $[2\Delta T, t_1]$ the one-iteration-ahead solution (say, Optimal trajectory 2), using $x_2$ as the new initial state. Then, denote by $k^2$ the feedback gain.

$$-B_{22}^{-1}\{f_u(t,x,u)^T K + B_{12}^T\}$$

and by $u^2$ the input corresponding to Optimal trajectory 2.

Step B-3: Apply to the plant the input $u^3$, $u^4$, $\cdots$.

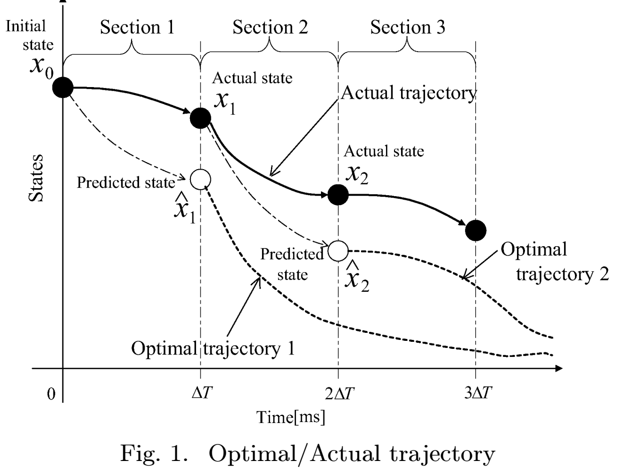

Fig. 1. Optimal/Actual trajectory

## 4. NUMERICAL SIMULATIONS

### A. Modeling

We demonstrate the effectiveness and practicability of our algorithmic controllers by applying them to the constrained optimal control problems, such as differential-driven mobile robot. The model of differential-driven mobile robot is shown in Fig.2. The torque of actuators are usually limited and thus we need to find a solution under the constrained conditions. System equation is given as follows.

$$\dot{\chi} = \frac{d\chi}{dt} = \frac{1}{2}\begin{pmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ 1/W & -1/W \end{pmatrix}\begin{pmatrix} u_R \\ u_L \end{pmatrix} \quad (11)$$

$$\chi = \begin{bmatrix} \chi_1 & \chi_2 & \chi_3 \end{bmatrix}^T = \begin{bmatrix} x & y & \theta \end{bmatrix}^T \quad (12)$$

Denote by $x$, $y$ and $\theta$ the position and angle of the mobile robot, by $u_L$ and $u_R$ the verosity of the left/right

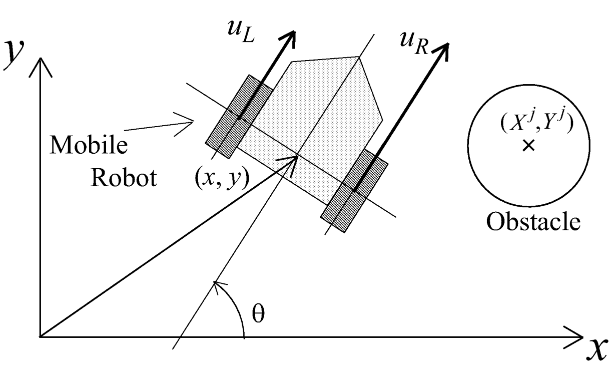

Fig. 2. Differential-driven mobile robot

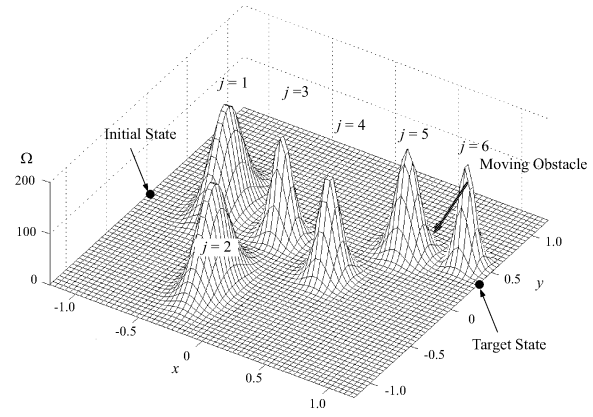

Fig. 3.   Obstacles

Table 1 Value of Obstacle Parameters

| $j$ | $(X^j, y^j)$ | $(\beta x^j, \beta y^j)$ | $R_S^j$ |
|---|---|---|---|
| 1 | $(-0.7, 0.2)$ | $(80, 20)$ | 200 |
| 2 | $(-0.3, -0.6)$ | $(80, 20)$ | 200 |
| 3 | $(-0.2, 0.1)$ | $(100, 100)$ | 200 |
| 4 | $(0.3, -0.1)$ | $(100, 70)$ | 200 |
| 5 | $(0.6, 0.4)$ | $(80, 80)$ | 200 |

wheel. The control input vector $u$ is given as $u = [v \ w]^{\mathrm{T}}$.

**B. Performance Index**

Basically, the performance index is as follows.

$$J = \xi^{\mathrm{T}} P \xi \mid_{t=t_1} + \frac{1}{2}\int_{t_0}^{t_1} (\xi^{\mathrm{T}} Q \xi + u^{\mathrm{T}} R u) dt \qquad (13)$$

$$\xi = [\ x - x_r \quad y - y_r \quad \sqrt{-\cos(\theta - \theta_r) + 1}\ ]^{\mathrm{T}} \qquad (14)$$

where $P$, $Q$ and $R$ are weight matrices.Denote by $x_r$, $y_r$ and $\theta_r$ the target position and angle of the robot. Here, setting the limitation on input vector by $u_{lim} = [0.25\mathrm{m/s} \ \ 0.25\mathrm{m/s}]^{\mathrm{T}}$, we introduce the following performance index.

$$\tilde{J} = J + \int_{t_0}^{t_1} r_L(0.25^2 - u_L^2)^{-1} + r_R(0.25^2 - u_R^2)^{-1} dt \qquad (15)$$

The weighting parameters are denoted by $r_L$ and $r_R$.

**C. Obstacles**

Here, we consider a problem where there exist obstacles

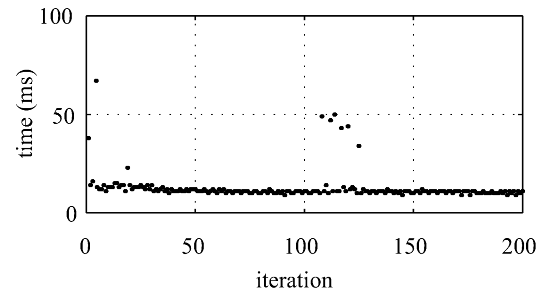

Fig. 4.   Calculation time

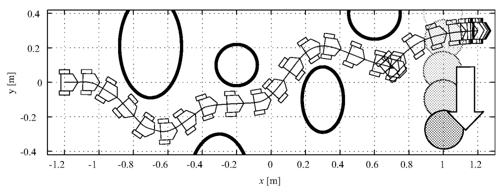

Fig. 5.   Trajectory of mobile robot

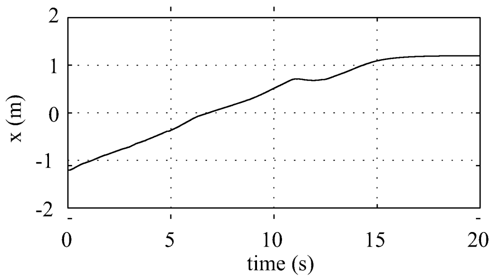

Fig. 6.   State trajectory $[x]$

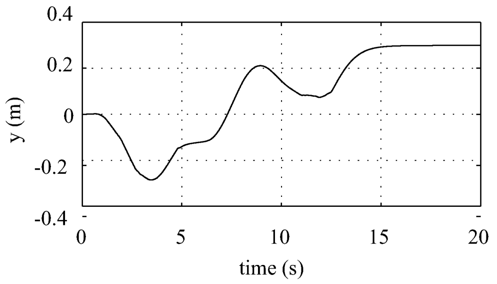

Fig. 7.   State trajectory $[y]$

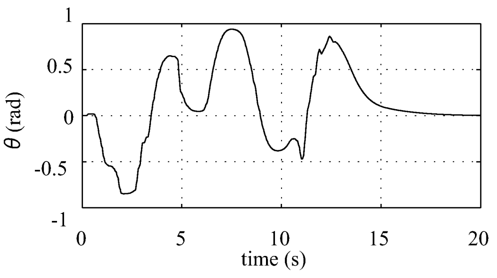

Fig. 8.   State trajectory $[\theta]$

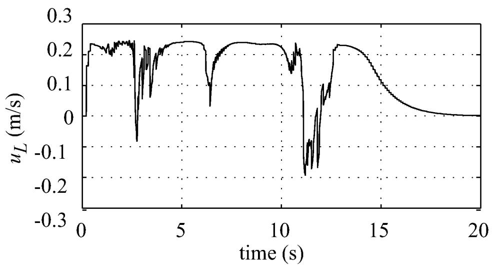

Fig. 9.   Control input $[u_L]$

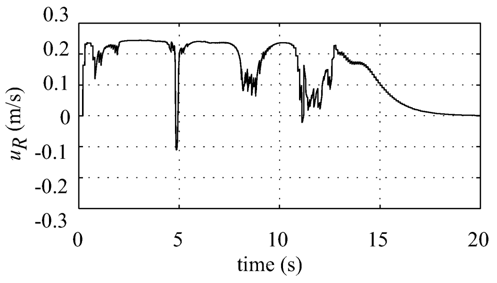

Fig. 10.   Control input $[u_R]$

in the movable region. Definition of the obstacles are as follows.

$$\Omega^j(x,y) = R_S^j \exp(-(x - X^j)^2\beta_x^j - (y - Y^j)^2\beta_y^j) \quad (16)$$

The position of obstacles are denoted by $X^j$ and $Y^j$. The shape of obstacle is characterized by $R_S^j$, $\beta_x^j$ and $\beta_y^j$ parameters. The suffix $j$ denotes the number of obstacles. We redefine performance index as follows.

$$\bar{J} = \tilde{J} + \int_{t_0}^{t_1} \sum_j \Omega^j dt \quad (17)$$

The calculation load increases, when the number of the obstacle increases. So, we only consider the obstacles which are close to the robot.

### D. Numerical Simulation Results

General PC is used for the simulation, and its specification is CPU: 2.53GHz, RAM: 512MByte. Multi-thread programming is used for simulator program.

Table 1 describes the condition of the obstacles for a numerical simulation. The interval of the right and left wheel is $2W = 0.128$[m]. Here, we set $t_1 = 1.0[s]$, and let the final time $t_1$ move as time goes by. The computation time for one-iteration-ahead solution is set by $\Delta T = 100$[ms]. The simulation results are shown in Figs. 4-10. The computation time for controllers in each section is sufficiently fast. Though the control input became discontinuous, good control performance is obtained. The mobile robot is able to avoid the moving obstacles skillfully. Therefore, it is possible to execute this algorithm in real time. The simulation results show the proposed controller works well in real time, even in the case of including moving obstacles.

## 5. HARDWARE EXPERIMENT

### A. Experimental System

We demonstrate the effectiveness and practicability of our algorithmic design method by applying it to the hardware experimental system. Here, the experimental system which we developed is described. The outline of developed experimental system is shown in Fig. 11. The condition of the mobile robot is observed using the outside fixed point camera. In this system, coordinate and angle of the robot can be measured by observing the light source of two LEDs of upper surface of the robot. The video signal of CCD camera is taken in using capture card on PC. The specification of the experimental equipment is shown in table 2 and the outline of developed robot is shown in Fig. 12.

### B. Experimental Results

We consider one obstacle which randomly moves. The initial state of the robot is measured and the target state is set to $[x_r \ y_r \ \theta_r] = [0 \ 0 \ 0]$. Other condition is same as the simulation. The experimental results are shown in Figs. 13-18. The mobile robot which we developed is able to avoid the moving obstacles skillfully. They show the proposed controller works well in real time, even in the case of including moving obstacles.

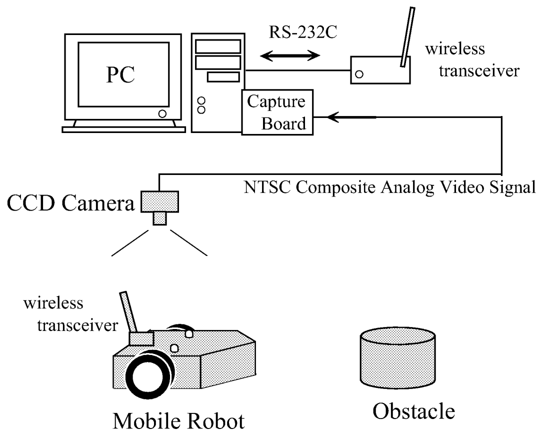

Fig. 11. Outline of developed experimental system

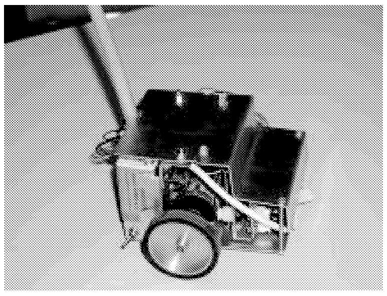

Fig. 12. Outline of developed robot

## 6. CONCLUSION

A design method for real-time calculation of constrained nonlinear optimal control problems is proposed. Our algorithmic method realizes dynamic optimization in real time for nonlinear systems. The proposed method is based on the optimal control algorithms. Whether or not the real-time computation works depends on how effective the algorithm is. Therefore, adopt the REB algorithm. It is applied to a differential-driven mobile robot, where there are input restrictions of actuators and there exist obstacles. The mobile robot which we developed is able to avoid moving obstacles skillfully. The numerical and experimental result show the proposed controller works well in real time, even in the case of including moving obstacles.

Table 2 Specifications of the experimental system

| ACTUATOR | | |
|---|---|---|
| DC MOTOR | Gear Ratio | 21 : 1 |
| | Voltage | 12 [V] |
| | Torque | 2.5 [Ncm] |
| | Speed | 80 [rpm] |
| COMMUNICATION | | |
| Wireless module | Baud rate | 9600 [bps] |
| | Packet Length | 16 [byte] |
| OBSERVING SYSTEM | | |
| CCD Camera | Total pixels | 0.25 [megapixels] |
| | Frame rate | 29.97 [fps] |

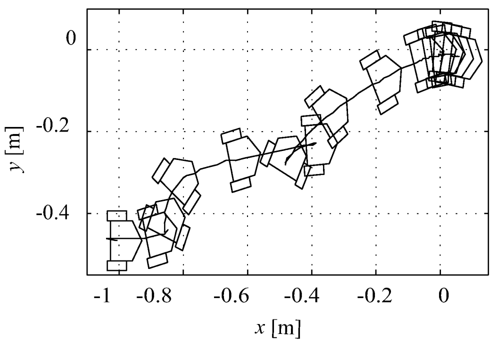

Fig. 13. Trajectory of mobile robot

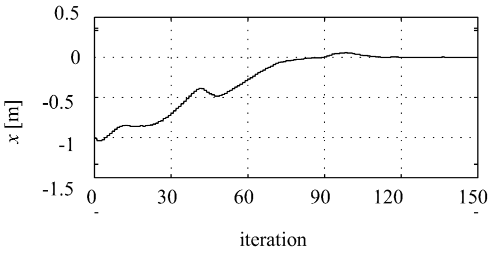

Fig. 14. State trajectory $[x]$

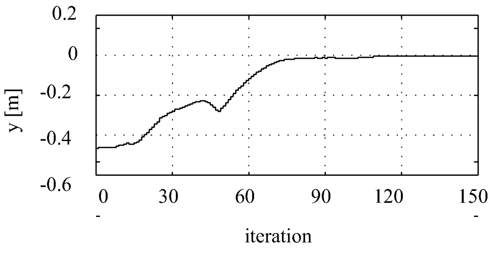

Fig. 15. State trajectory $[y]$

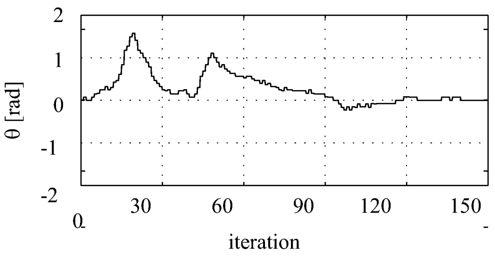

Fig. 16. State trajectory $[\theta]$

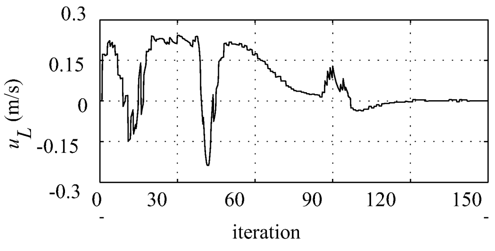

Fig. 17. Control input $[u_L]$

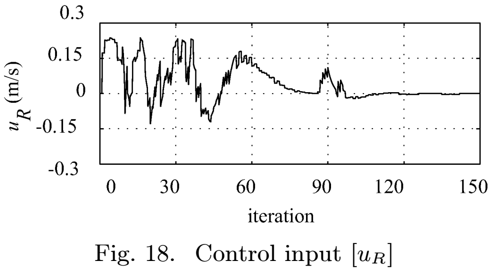

Fig. 18. Control input $[u_R]$

## References

[1] D. H. Jacobson, and D. Q. Mayne, *Differential Dynamic Programming*, Elsevier, 1970.

[2] N. B. Nedeljkovic, "New Algorithms for Unconstrained Nonlinear Optimal Control Problems," *IIEEE Transactions on Automatic Control*, vol. AC26, pp. 868–884, April 1981.

[3] J. Imae and R. Torisu, "A Riccati-Equation Based Algorithm for Nonlinear Optimal Control Problems," *Proc. of the 37th IEEE CDC*, pp. 4422–4427, 1998.

[4] W. H. Kwon and A. E. Pearson, "A Modified Quadratic Cost Problem and Feedback Stabilization of a Linear System," *IEEE Transactions on Automatic Control*, vol. 22-5, pp. 838–842, 1986.

[5] D. Q. Mayne and H. Michalska, "Receding Horizon Control of Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. 35-7, pp. 814–824, 1990.

[6] F. Allgöwer and A. Zheng, editors. *Nonlinear Model Predictive Control*, Birkhäuser, 2000.

[7] T. Otsuka, "A Continuation/GMRES Method for Fast Computation of Nonlinear Receding Horizon Control," *Automatica*, vol. 40, pp. 563–574, 2004.

[8] C. W. Merriam III, "A Computational Method for Feedback Control Optimization.," *Information and Control*, vol. 8, pp. 215–232, 1965.

[9] D. M. Murray, *Differential Dynamic Programming for the Efficient Solution. of Optimal Control Problems*, University of Arizona, PhD Thesis, 1978.

[10] J. Imae, L. Irlicht, G. Obinata and J. B. Moore, "Enhancing Optimal Controllers via Techniques. From Robust and Adaptive Control," *International Journal of Adaptive Control and Signal Processing*, vol. 6, pp. 413–429, 1992.

[11] J. R. Cloutier, C.N. D'Souza and C.P. Mracek, "Nonlinear Regulation and Nonlinear $H_\infty$ Control via the State-Dependent Riccati Equation Technique, Part 1&2.," *Proc. 1st international conference on nonlinear problems in aviation and aerospace*, 1996.

[12] P. Dyer, and S. R. McReynolds, *The Computation and Theory of Optimal Control*, Academic Press, 1970.

[13] T. E. Bullock, and G. F. Franklin, "A Modified Quadratic Cost Problem and Feedback Stabilization of a Linear System,' *IEEE Transactions on Automatic Control*, vol. 12-6, pp. 666–673, 1967.

[14] T. Kobayashi, J. Imae, M. Magono, K. Yoshimizu and G. Zhai, "Real-Time Optimization for Nonlinear Systems Using Algorithmic Control,' The 16th IFAC World Congress, (in press) (2005)