Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Algorithmic control for real-time optimization of nonlinear systems: Simulations and experiments

Joe Imae, Kazutaka Yoshimizu, Tomoaki Kobayashi, and Guisheng Zhai
Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka 599-8531, JAPAN

Abstract: A new method for real-time calculation of nonlinear optimal control problems is proposed. The proposed method is based on the iterative algorithms for obtaining the numerical solutions of optimal control problems. In this paper, we adopt the Riccati-equation based algorithm, which is one of the iterative algorithms, and demonstrate that the proposed 'algorithmic controller' is applicable to the Van der Pol problem, the crane problem, and the swing-up problem of an inverted pendulum by means of simulations and experiments.

1. Introduction

The design methods for real-time calculation of unconstrained/constrained nonlinear control problems have gained much attention in a variety of real-world applications. Among them, Model Predictive Control (MPC) and State-Dependent-Riccati-Equation (SDRE) techniques are getting their popularity [1], [2], [3].

In the computational field of control problems, iterative approaches are popular in finding numerical solutions. It is known, however, that such approaches are computationally expensive, and that they are not suitable in the situation of real-time control. In this paper, we propose a new idea in order to decrease such computational burden. Based on such a new idea, we focus on the iterative computational techniques for numerical solutions, and attack the big issue of real-time calculation of nonlinear optimal control problems. Also, we attack the constrained optimal control problems.

The outline of the paper is as follows. In Section 2, we describe the unconstrained/constrained optimal control problems and show how the constrained problems are converted into unconstrained ones by means of penalty function methods. Also, a computational method for numerical solutions of optimal control problems is given with its convergence property. In Section 3, based on the computational method of optimal control problems, our algorithmic design method is proposed. In Section 4, some simulations and experiments are given in order to demonstrate the effectiveness and practicability of our approach, where the Van der Pol problem, the crane problem, and the swing-up problem of an inverted pendulum are adopted as the design examples.

2. Optimal control problem

2.1 Formulation

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

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

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

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

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 control interval $[t_0, t_1]$. Based on the problem formulation (1) to (3), we describe our on-line computational design method, that is to say, algorithmic design method. Here, denote the state variable by $x(t) = [x_1(t), \cdots, x_n(t)]^T \in R^n$, and the input variable by $u(t) = [u_1(t), \cdots, u_r(t)]^T \in R^r$.

It is worthwhile to note that, even in the case of terminal conditions and state/control constraints being given, we could transform the constrained ones into the unconstrained ones by means of penalty function methods [4]. Take the following constraints for instance:

$$\begin{aligned} &S_i(x(t_1)) = 0, \qquad i = 1, \cdots, p_1 \\ &S_i(x(t), u(t)) = 0, \qquad i = p_1 + 1, \cdots, p_2 \\ &S_i(x(t), u(t)) \geq 0, \qquad i = p_2 + 1, \cdots, p \end{aligned} \tag{4}$$

By the penalty function methods, we can formulate the unconstrained optimal control problem as follows.

$$\begin{aligned} &\dot{x}(t) = f(t, x(t), u(t)) \\ &x(t_0) = x_0 \in R^n \\ &J = G(x(t_1)) + \textstyle\sum_{i=1}^{p_1} r_i S_i(x(t_1))^2 \\ &\quad + \int_{t_0}^{t_1} \{L(t, x(t), u(t)) + \textstyle\sum_{i=p_1+1}^{p_2} r_i S_i(x(t), u(t))^2 \\ &\quad + \textstyle\sum_{i=p_2+1}^{p} [r_i / S_i(x(t), u(t))]\} dt \end{aligned} \tag{5}$$

where $r_i (i = 1, \cdots, p_2)$ are exterior penalty parameters and $r_i (i = p_2 + 1, \cdots, p)$ are interior ones. Note that the terminal conditions and state/control constraints (4) are included in the performance index (5) with penalty parameters.

Now, we give some preliminaries. Whether or not the algorithmic design method succeeds would depend on how effective the algorithm is in iteratively 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 are known to be reliable and effective in searching numerical solutions. One of the characteristics is the use of feedback structure in the

0-7803-9568-9/05/$20.00 ©2005 IEEE

calculation process for numerical solutions. Details are given later.

2.2 REB algorithm [5]

2.2.1 Assumptions

Focusing on 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 [6]-[12].

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

$$\|x\| = \max |x(t)| \ \ for\ x \in AC^j \ , \ t \in [t_0\ t_1]$$

$$\|y\| = ess\ \sup |y(t)| \ \ for \ \ y \in L_\infty^j \ , \ t \in [t_0\ t_1]$$

where the vertical bars are used to denote Euclidean norms for vectors.

Now, we make some assumptions.

(1) $G:R^n\to R^1$, $f: R^1\times R^n\times R^r\to R^n$, $L: R^1\times R^n \times R^r \to R^1$ 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.

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

$$|f(t,x,u)|\le M_1(|x|+1)$$

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

2.2.2 Algorithm

<Step 0> Let $\beta\in(0,1)$ and $M_2\in(0,1)$. Select arbitrarily an initial input $u^0\in L_\infty^r$.

<Step 1> $i=0$.

<Step 2> Calculate $x^i(t)$ with $u^i(t)$ from the equation (1).

<Step 3> Select $A^i\in R^{n\times n}, B_{11}^i\in L_\infty^{n\times n}$, $B_{12}^i\in L_\infty^{n\times r}$, and $B_{22}^i\in L_\infty^{r\times r}$, so that Kalman's sufficient conditions for the boundedness of Riccati solutions [9, p.36] hold, that is, for almost all $t\in[t_0,t_1]$,

$$A^i\ge 0$$

$$B_{22}^i(t)>0,\ B_{11}^i(t)-B_{12}^i(t)\left(B_{22}^i(t)\right)^{-1}\left(B_{12}^i(t)\right)^T\ge 0$$

where A^i, B_{11}^i and B_{22}^i are symmetric and $(\bullet)^T$ means the transpose of vectors and matrices.

We solve $\delta x^i(t), K^i(t), r^i(t)$ from (6), (7), and (8) below,

$$\begin{aligned}\delta\dot{x}(t) = &\Big\{ f_x(t,x^i,u^i)+f_u(t,x^i,u^i)\left(B_{22}^i\right)^{-1}\\ &\times\left(f_u{}^T(t,x^i,u^i)K(t)-\left(B_{12}^i\right)^T\right)\Big\}\delta x(t)\\ &+f_u(t,x^i,u^i)\left(B_{22}^i\right)^{-1}\Big(f_u{}^T(t,x^i,u^i)\ r(t)\\ &-L_u{}^T(t,x^i,u^i)\Big)\ ,\ \delta x(t_0)=0 \end{aligned} \quad (6)$$

$$\begin{aligned}\dot{K}(t) = &-K(t)f_x(t,x^i,u^i)-f_x{}^T(t,x^i,u^i)\ K(t)\\ &+B_{11}^i+\left(K(t)f_u(t,x^i,u^i)-B_{12}^i\right)\left(B_{22}^i\right)^{-1}\\ &\times\Big(\left(B_{12}^i\right)^T-f_u{}^T(t,x^i,u^i)\ K(t)\Big),\ K(t_1)=-A^i \end{aligned} \quad (7)$$

$$\begin{aligned}\dot{r}(t) = &-f_x{}^T(t,x^i,u^i)r(t)+L_x{}^T(t,x^i,u^i)\\ &+\left(B_{12}^i-K(t)f_u(t,x^i,u^i)\right)\left(B_{22}^i\right)^{-1}(-L_u{}^T(t,x^i,u^i)\\ &+f_u{}^T(t,x^i,u^i)\ r(t)\ ),\ r(t_1)=-\mathrm{G}(x(\mathrm{t}_1)) \end{aligned} \quad (8)$$

and determine δu^i from the following.

$$\begin{aligned}\delta u^i(t) = &\left(B_{22}^i\right)^{-1}\Big[\Big(\Big(f_u{}^T(t,x^i,u^i)\ K^i(t)\\ &-\left(B_{12}^i\right)^{\mathrm{T}}\Big)\delta x^i+f_u{}^T(t,x^i,u^i)\ r^i(t)\\ &-L_u{}^T(t,x^i,u^i)\Big] \end{aligned} \quad (9)$$

<Step 4> Determine $(\tilde{x}^i,\tilde{u}^i)$ satisfying

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

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

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

where

$$\begin{aligned}H^i(t,\delta x,\delta u,p) = &-\Big\{L_x(t,x^i,u^i)\delta x+L_u(t,x^i,u^i)\delta u\\ &+\frac{1}{2}\left[\delta x^T B_{11}^i\delta x+2\delta x^T B_{12}^i\delta u+\delta u^T B_{22}^i\delta u\ \right]\Big\}\\ &+p^T\left[f_x(t,x^i,u^i)\delta x\ +f_u(t,x^i,u^i)\delta u\ \right]\end{aligned}$$

and p^i is a solution of the followings.

$$\dot{p}(t)=-f_x{}^T(t,x^i,u^i)\ p(t)+L_x{}^T(t,x^i,u^i)\ ,$$

$$p(t_1)=-G_x{}^T(x(t_1))$$

<Step 5> $\alpha_i=1$.

<Step 6> Set

$$u^{i+1}(t)=u^i(t)+\alpha_i\delta u^i(t)+\alpha_i^2\left(\tilde{u}^i(t)-u^i(t)-\delta u^i(t)\right)$$

If (10) holds, go to Step 7. Otherwise, set $\alpha_i=\beta\alpha_i$ and repeat Step 6.

$$\begin{aligned}J(u^{i+1})-J(u^i)\le\alpha_i M_2\{&G(x_i(t_1))\delta x(t_1)\\ &+\int_{t_0}^{t_1}\left[L_x(t,x^i,u^i)\delta x^i+L_u(t,x^i,u^i)\delta u^i\ \right]dt\ \} \end{aligned} \quad (10)$$

<Step 7> Set $i = i+1$, and go to Step 3. Repeat Step 3 to Step 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. Proof is omitted here. See [5] for more details.

Theorem (Convergence)

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

$$|\tilde{u}^i| \le M_4 \text{ and } |\delta u^i| \le M_4, i = 0,1,2\cdots$$

and also suppose that exist $\bar{A} \in R^{n\times n}, \bar{B}_{11} \in L_\infty^{n\times n}$, $\bar{B}_{12} \in L_\infty^{n\times r}, \bar{B}_{22} \in L_\infty^{r\times r}, \bar{u} \in L_\infty^r$ and a sequence $N \subset (0,1,2,3,\cdots)$ such that

$$\bar{A} \ge 0$$

$$\bar{B}_{11}(t) > 0,\ \bar{B}_{11}(t) - \bar{B}_{12}(t)\left(\bar{B}_{22}(t)\right)^{-1}\left(\bar{B}_{12}(t)\right)^T \ge 0 \quad \text{for almost all } t \in [t_0, t_1]$$

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

$$\lim_{i\in N} B_{11}^i(t) \ \left(\text{resp.}, B_{12}^i(t), B_{22}^i(t)\right) = \bar{B}_{11}(t) \ \left(\text{resp.}, \bar{B}_{12}(t), \bar{B}_{22}(t)\right)$$

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

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

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

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

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.

$$\dot{p}(t) = -f_x{}^T(t, \bar{x}, \bar{u})\ p(t) + L_x{}^T(t,x,u)$$

$$p(t_1) = -G_x{}^T(\bar{x}(t_1))$$

3. Algorithmic design

We propose a new design tool for real-time optimization of nonlinear systems. The key idea is very simple. Roughly speaking, all we have to do is to proceed with the following procedure in the real-time situation. That is, we carry out the REB algorithm through a few iterations, pick up the calculated control and apply it to the system, and then we continuously carry out the REB algorithm through a few iterations, pick up the calculated control and apply it to the system, and so on. This means that the number of iterations increases as time goes by. That is, it could be expected to eventually reach the possible optimal solutions after sufficiently large number of iterations. More detailed explanations are given in the following description.

From a practical point of view, the calculation time for a-few-iteration-ahead solution is assumed to be equal to ΔT [ms] (or less than ΔT). The calculation time ΔT plays a key role in our design method. We here describe how the proposed controller works. See also Figure 1.

Algorithm

<Step1> Measure an actual state x_0, and select arbitrarily an initial input u^0. Set the unit of calculation time as ΔT [ms], and apply the input u^0 to the system over the interval of the first unit time of calculation (say, Section 1). In Section 1, we proceed with two kinds of calculations: One is to predict the one-unit-time-ahead state $\hat{x}_1$ through the system equation (1) with the initial state x_0, and the other is to calculate the a-few-iteration-ahead solution over $[\Delta T, t_1]$ with the updated initial state $\hat{x}_1$ (say, Optimal trajectory 1). Then, denote by k^1 a feedback gain

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

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

<Step2> Measure the actual state x_1, and apply the input u^1 together with the feedback gain k^1 to the system over the interval of the second unit time of calculation (say, Section 2). In Section 2, we proceed with two kinds of calculations: One is to predict the one-unit-time-ahead state $\hat{x}_2$ through the system equation (1) with the state x_1, and the other is to calculate the a-few-iteration-ahead solution over $[2\Delta T, t_1]$ with the updated state x_2 (say, Optimal trajectory 2). Then, denote by k^2 a feedback gain.

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

and by u^2 the input associated with Optimal trajectory 2.

<Step3> Apply to the system the input $u^3, u^4, u^5, \cdots$.

The real-time control $u^1, u^2, u^3, \cdots$ with $k^1, k^2, k^3, \cdots$ is called 'algorithmic control', and the similar idea is given for the unconstrained optimal control problems in [13].

4. Simulations and experiments

We demonstrate the effectiveness and practicability of our algorithmic control by applying them to the Van der Pol problem, the crane problem, and the swing-up problem of the inverted pendulum

4.1 Simulation results

We deal with the Van der Pol Oscillator. System equation, initial condition, and performance index are as follows.

$$\dot{x}_1(t) = \left(1 - x_2^2(t)\right)x_1(t) - x_2(t) + u$$
$$\dot{x}_2(t) = x_1(t) \qquad (11)$$
$$x_1(0) = 0, x_2(0) = 1$$

$$J(u) = \frac{1}{2}\int_0^\infty \left(x_1^2(t) + x_2^2(t) + u^2(t)\right)dt \qquad (12)$$

Our goal is to find a real-time optimal controller in the algorithmic fashion. We apply the algorithmic design method to this oscillation problem.

In this example, $t_1 = \infty$ is given. However, from the computational point of view, it is impossible to find a numerical solution with the terminal time being infinity. So, we here set $t_1 = 10$, which would be sufficiently large, and besides let the final time t_1 move as time goes by. The integration steps of differential equations are chosen to be 1000, and the initial input is chosen as $u^0(t) = 0$. The computation time for a-few-iteration-ahead solution is set by $\Delta T = 100$ [ms].

The CPU time is given in Figure 2, where the computation time for a-few-iteration-ahead solution has turned out less than 100[ms]. This implies that the algorithmic design method is implementable in terms of real-time optimization. Note that ten iterations are carried out within the unit time of calculation in simulation. Figure 3 shows the computed control input compared with the numerical optimal solution. Surprisingly, those are so close.

4.2 Experimental results

We deal with the swing control of the crane system, and swing-up control of the inverted pendulum. We here adopt the same experimental equipments for the crane system and inverted pendulum system. See Fig.4.

System equation is given as follows. Denote by r the position of the cart, by θ the angular of the pendulum, and by v the input voltage to the DC motor. Therefore, the state variable x is given as

$$x = \{x_1\ x_2\ x_3\ x_4\}^T = \{r\ \theta\ \dot{r}\ \dot{\theta}\}^T$$

and the input variable u is given as $u = v$. See Figure 5.

$$\dot{x} = Ax + Bu$$
$$A = \frac{1}{\alpha_1}\begin{bmatrix} 0 & 0 & \alpha_1 & 0 \\ 0 & 0 & 0 & \alpha_1 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix}, \quad B = \frac{1}{\alpha_1}\begin{bmatrix} 0 \\ 0 \\ b_3 \\ b_4 \end{bmatrix}$$

where

$$a_{32} = -(ml)^2 g \cos x_2 \frac{\sin x_2}{x_2}$$
$$a_{33} = -(J + ml^2)C_c$$
$$a_{34} = -(J + ml^2)mlx_4 \sin x_2 - mlC_p \cos x_2$$
$$a_{42} = -(M + m)mgl\frac{\sin x_2}{x_2}$$
$$a_{43} = -mlC_c x_3 \cos x_2$$
$$a_{44} = -(ml)^2 x_4 \cos x_2 \sin x_2 - (M + m)C_p$$
$$b_3 = k(J + ml^2)$$
$$b_4 = kml \cos x_2$$
$$\alpha_1 = MJ - (ml)^2 \cos^2 x_2$$

and the values of parameters are given in Table 1. Note that the input constraint is imposed. The torque of DC motors is usually limited and we have to find a solution under the constrained situation.

4.2.1 Swing control of the crane

First, we introduce the following performance index.

$$J(u) = \frac{1}{2}\int_0^{t_1} \left(100 \cdot x_1^2(t) + 10x_2^2(t) + x_3^2(t) + 10x_4^2(t) + u^2(t) + \frac{0.0001}{24^2 - u^2}\right)dt$$

Here, the term of $(24^2 - u^2)^{-1}$ plays a role to keep the input away from the boundary of the practical limitation of the DC motor. The limitation of the allowable input voltage is 24 volts.

We set $\Delta T = 100$ [ms] and $t_1 = 1$. The integration steps of differential equations are set to 100 steps, and the input $u^0(t) = 0$ is adopted as an arbitrarily chosen initial one. It would be important to note that the terminal time t_1 is not fixed, that is, the terminal time t_1 moves as the time goes by.

The CPU time for one-iteration-ahead solution has turned out less than $\Delta T = 100$ [ms]. This means that such algorithmic design methods are implementable in terms of real-time optimization. The behavior of the angle with $x_2(0) \cong -1.2$ is shown in Figure 6, where the angle of the pendulum quickly goes to zero in about 3 minutes.

4.2.2 Swing-up control of the inverted pendulum

The following performance index is given.

$$J(x,u) = 2.5 \cdot x_1^2(t) + 100 \cdot \sin^2(0.5 \cdot x_2(t)) + 0.01 \cdot x_3^2(t) + 0.5 \cdot x_4^2(t) + (1/2)\int_{t_0}^{t_1} \left(5 \cdot x_1^2(t) + 5 \cdot \sin^2(0.5 \cdot x_2(t)) + 0.001 \cdot x_3^2(t) + 0.15 \cdot x_4^2(t) + 0.4 \cdot u^2(t) + 0.1 \cdot (24^2 - u^2)^{-1} \right)dt$$

Here, the term of $(24^2 - u^2)^{-1}$ is related to the limitation of the voltage of the DC motor as stated before. We set $\Delta T = 100$ [ms] and $t_1 = 1.5$. The integration steps of differential equations are set to 150 steps, and the input $u^0(t) = 0$ is adopted as an initial one arbitrarily chosen. Note that five iterations are carried out within the unit time of calculation in experiment.

Figure 7 tells us that the swing-up control of the inverted pendulum has succeeded even under the input constraint. The CPU time for the calculation of a-few-iteration-ahead solution has turned out less than $\Delta T = 100$ [ms]. The behaviors of states and input with $x_2(0) = -3.14$ are shown in Figure 8 and 9.

5. Conclusion

A design method for real-time calculation of unconstrained/constrained nonlinear optimal control problems has been proposed. The proposed method is based on the iterative computational techniques for the optimal control problems. It has been demonstrated that the algorithmic controller is applicable to the Van der Pol problem, the swing control problem of the crane system, and the swing-up control problem of the inverted pendulum.

References

[1] F. Allgowet, and A. Zhenget, Nonlinear Model Predictive Control, Birkhauser (2000)

[2] T. Otsuka, A continuation/GMRES method for fast computation of nonlinear receding horizon control, Automatica, Vol. 40, pp. 563-574 (2004)

[3] J. R. Cloutier, C. N. D'Souza, and C. P. Mracek, Nonlinear regulation and nonlinear H_∞ control via the state-dependent Riccati equation technique, Part 1&2, Proc. 1st International Conference on Nonlinear Problems in Aviation and Aerospace (1996)

[4] J. Imae, and K. Inoue, Unified approach to computational methods of nonlinear optimal control problems with possible jumps in states, Proc. of the 2002 ACC, pp.4571-4576 (2002)

[5] 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)

[6] C. W. Merriam III, A computational method for feedback control optimization, Information and Control, Vol.8, pp.215-232 (1965)

[7] T. E. Bullock, and G. F. Franklin, A second-order feedback method for optimal control computations, IEEE Trans. Automatic Control, AC-12-6, pp.666-673 (1967)

[8] P. Dyer, and S. R. McReynolds, The Computation and Theory of Optimal Control, Academic Press (1970)

[9] D. H. Jacobson, and D. Q. Mayne, Differential Dynamic Programming, American Elsevier, New York (1970)

[10] J. Imae, L. Irlicht, G. Obinata and J. B. Moore, Enhancing optimal controllers via techniques. from robust and adaptive control, Int. J. of Adaptive Control and Signal Processing, Vol.6, pp.413-429 (1992)

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

[12] N. B. Nedeljkovic, New algorithms for unconstrained nonlinear optimal control problems, IEEE Trans. Automatic Control, AC-26, pp.868-884 (1981)

[13] T. Kobayashi, M. Magono, J. Imae, Y. Yoshimizu, and G. Zhai, Real-time optimization for nonlinear systems using algorithmic control, 16th IFAC World Congress (2005)

Table 1 Parameters of experimental equipments

M	mass of cart	0.395 kg
k_f	torque coefficient	1.4 N/V
J	moment of inertia	3.02×10^{-3} kgm^2
l	length of pendulum	0.301 m
m	mass of pendulum	0.100 kg
C_c	friction coefficient between rail and cart	12.9 kg/s
C_p	friction coefficient between cart and pendulum	1.55×10^{-5} kgm^2/s
g	gravitational acceleration	9.806 m/s^2

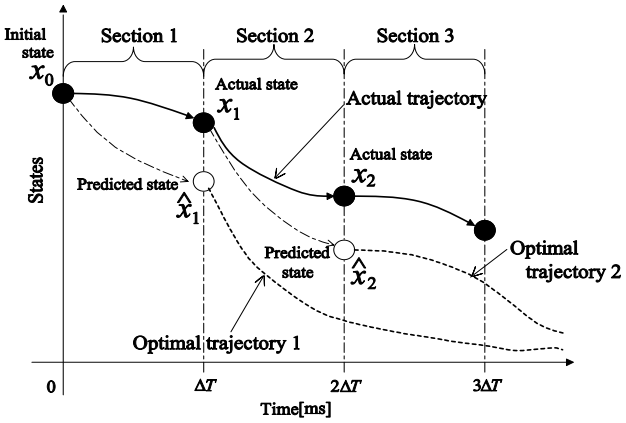


Figure 1 Optimal/Actual trajectories

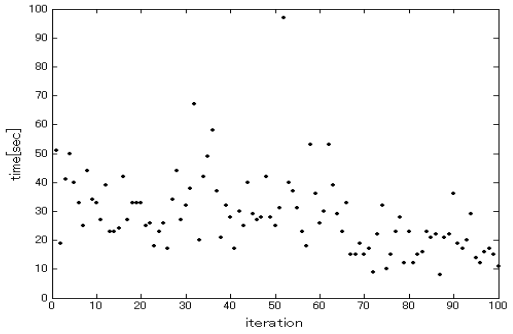


Figure 2 CPU time

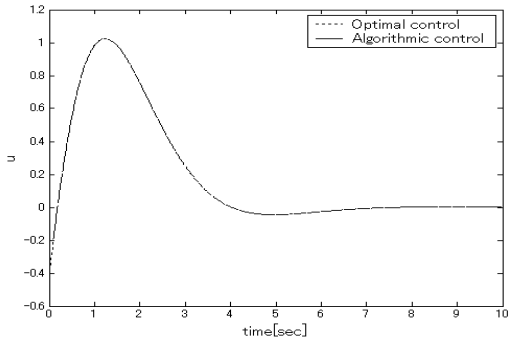


Figure 3 Optimal/algorithmic controls

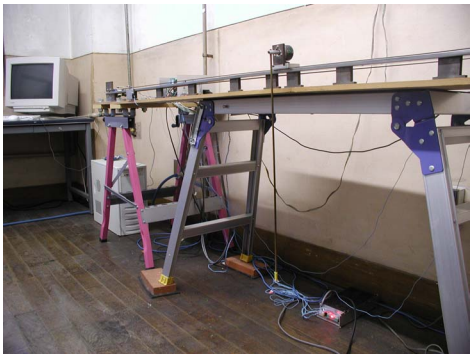


Figure 4 Experimental equipments

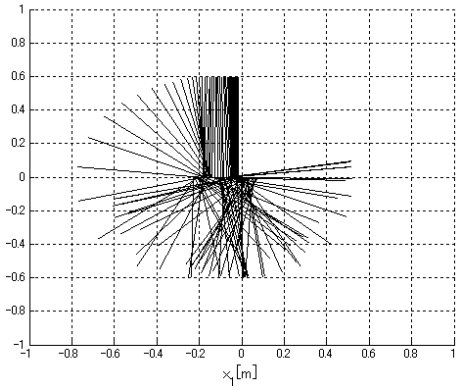
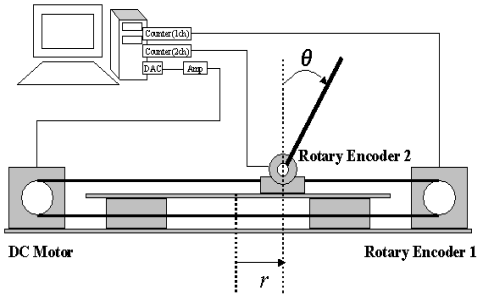
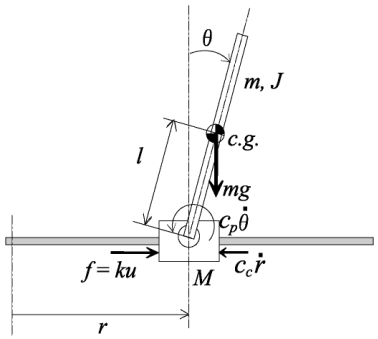


Figure 7 Experimental behavior of the pendulum



(a) wholel view



(b) Pendulum

Figure 5 Schematic representation of experimental equipments

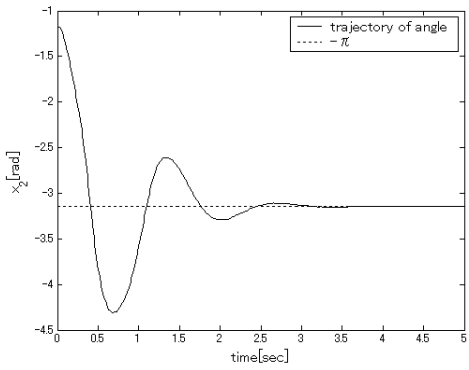


Figure 6 State x_2

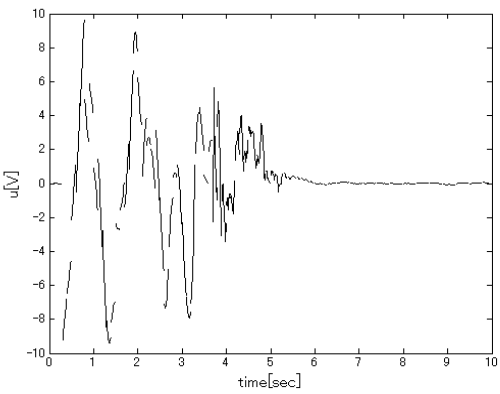


Figure 8 History of control input

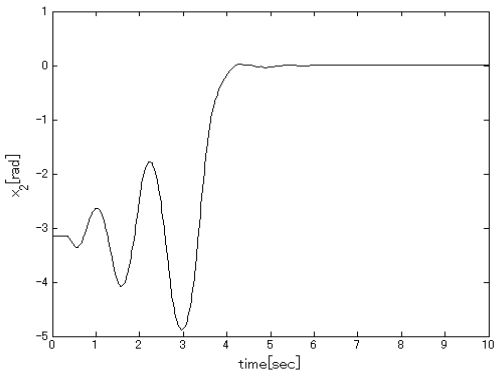


Figure 9 History of state x_2