Continuation/GMRES Method for Fast Algorithm of Nonlinear Receding Horizon Control

Toshiyuki Ohtsuka
Department of Computer-Controlled Mechanical Systems
Osaka University, Osaka 565-0871, Japan
ohtsuka@mech.eng.osaka-u.ac.jp

Abstract

This paper proposes a fast algorithm for nonlinear receding horizon control. The control input is updated by a differential equation to trace the solution of an associated two-point boundary-value problem. A linear equation involved in the differential equation is solved by the generalized minimum residual (GMRES) method, one of the Krylov subspace methods, with Jacobians approximated by forward differences. The error in the entire algorithm is analyzed and is shown to be bounded under mild conditions. The proposed algorithm is applied to a two-link arm whose dynamics is highly nonlinear.

1 Introduction

The main concern of this paper is numerical algorithms for nonlinear receding horizon control (model predictive control). As capability of digital computers grows, nonlinear receding horizon control is applied successfully to chemical industry [1], where the sampling interval is sufficiently large, e.g, several ten seconds or larger, and a successive approximation algorithm can be executed to solve the problem within the sampling interval. However, successive approximation is computationally expensive and is not suitable for mechanical systems controlled with a sampling interval in the order of milliseconds. Although an approximate algorithm of nonlinear receding horizon control can be obtained explicitly through use of the Taylor expansion [2], the length of the horizon and the form of the performance index are restricted in such an algorithm. Efficient numerical algorithms should be explored to broaden applications of receding horizon control.

This paper proposes a real-time algorithm of nonlinear receding horizon control by combining the continuation method with a fast algorithm for linear equations. The problem is discretized over the horizon first, and then a differential equation to update the sequence of control inputs is obtained through use of the continuation method. Since that differential equation involves a large linear equation, we need a fast algorithm to solve it. We employ the GMRES (Generalized Minimum RESidual) method [3] to solve the linear equation. The error in the entire algorithm is analyzed and is shown to be bounded under some conditions.

The proposed algorithm is applied to a two-link arm in order to examine the computational time. Simulation results show that nonlinear receding horizon control is possible in real time for the highly nonlinear system with the proposed algorithm.

Throughout the paper, the symbol $\|\cdot\|$ denotes the Euclidean norm for a vector and the induced norm (maximum singular value) for a matrix, respectively. For a vector x_0 and a positive scalar ε, let $\mathcal{B}(x_0, \varepsilon) = \{x : \|x - x_0\| \leq \varepsilon\}$.

2 Nonlinear Receding Horizon Control Problem

This section briefly summarizes the nonlinear receding horizon control problem. We consider a general nonlinear system governed by a state equation

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

where $x(t) \in \mathbf{R}^n$ denotes the state vector, $u(t) \in \mathbf{R}^{m_u}$ the control input vector, and $p(t) \in \mathbf{R}^{m_p}$ the vector of given time-dependent parameters, respectively. The control input at each time t is determined so as to minimize a performance index with a receding horizon:

$$\begin{aligned} J &= \varphi(x(t+T), p(t+T)) \\ &\quad + \int_t^{t+T} L(x(t'), u(t'), p(t'))dt'. \end{aligned}$$

Equality constraints are also imposed in general as:

$$C(x(t), u(t), p(t)) = 0,$$

where C is an m_c dimensional vector-valued function. An inequality constraint can be converted to an equality constraint with a dummy input (slack variable). Therefore, the present problem setup can deal with various performance indexes and constraints.

At each time t, we have to find the optimal trajectory starting from the present state $x(t)$. To this end, we divide the horizon into N steps and discretize the optimal control problem with the forward difference as follows:

$$x_{i+1}^*(t) = x_i^*(t) + f(x_i^*(t), u_i^*(t), p_i^*(t))\Delta\tau, \quad (1)$$
$$x_0^*(t) = x(t), \quad (2)$$
$$C(x_i^*(t), u_i^*(t), p_i^*(t)) = 0, \quad (3)$$
$$J = \varphi(x_N^*(t), p_N^*(t)) \quad (4)$$
$$+ \sum_{i=0}^{N-1} L(x_i^*(t), u_i^*(t), p_i^*(t))\Delta\tau, \quad (5)$$

where $\Delta\tau := T/N$, $x_i^*(t)$ corresponds to the state at time $t + i\Delta\tau$ on the optimal trajectory starting from $x(t)$, and $p_i^*(t)$ is given by $p(t+i\Delta\tau)$. Since the horizon T depends on time t in general, so does $\Delta\tau$. Any other one-step schemes may also be employed to discretize the problem. Note that although the optimal control problem on the horizon is discretized, its dependence on t remains continuous. Given the initial state of the discretized problem, $x_0^*(t) = x(t)$, the control sequence $\{u_i^*(t)\}_{i=0}^{N-1}$ is optimized at each time t. The actual control input to the system is given by $u(t) = u_0^*(t)$.

Let H denote the Hamiltonian defined by

$$H(x, \lambda, u, \mu, p) := L(x, u, p) + \lambda^T f(x, u, p) + \mu^T C(x, u, p).$$

where $\lambda \in \mathbf{R}^n$ denotes the costate, and $\mu \in \mathbf{R}^{m_c}$ denotes the Lagrange multiplier associated with the equality constraint. The first-order necessary conditions for the sequences of optimal control $\{u_i^*(t)\}_{i=0}^{N-1}$, multiplier $\{\mu_i^*(t)\}_{i=0}^{N-1}$ and costate $\{\lambda_i^*(t)\}_{i=0}^{N}$ are obtained by the calculus of variation as

$$H_u(x_i^*(t), \lambda_{i+1}^*(t), u_i^*(t), \mu_i^*(t), p_i^*(t)) = 0, \quad (6)$$
$$\lambda_i^*(t) = \lambda_{i+1}^*(t) \quad (7)$$
$$+ H_x^T(x_i^*(t), \lambda_{i+1}^*(t), u_i^*(t), \mu_i^*(t), p_i^*(t))\Delta\tau, \quad (8)$$
$$\lambda_N^*(t) = \varphi_x^T(x_N^*(t), p_N^*(t)), \quad (9)$$

The sequences of the optimal control $\{u_i^*(t)\}_{i=0}^{N-1}$ and the multiplier $\{\mu_i^*(t)\}_{i=0}^{N-1}$ have to satisfy Eqs. (1) - (3), (6) - (9), which define a TPBVP (Two-Point Boundary-Value Problem) for the discretized optimal control problem. The solution of the discretized problem converges to the solution of the continuous problem as $N \to \infty$ under mild conditions [4].

We define a vector of the inputs and multipliers as

$$U(t) := [u_0^{*T}(t)\ \mu_0^{*T}(t)\ u_1^{*T}(t)\ \mu_1^{*T}(t)\ \ldots\ u_{N-1}^{*T}(t)\ \mu_{N-1}^{*T}(t)]^T \in \mathbf{R}^{mN},$$

where $m := m_u + m_c$. We also define a projection $P_0 : \mathbf{R}^{mN} \to \mathbf{R}^{m_u}$ as

$$P_0(U) := u_0^*.$$

For a given $U(t)$ and $x(t)$, $\{x_i^*(t)\}_{i=0}^{N}$ are calculated recursively by Eqs. (1) and (2), and then, $\{\lambda_i^*(t)\}_{i=0}^{N}$ are also calculated recursively from $i = N$ to $i = 0$ by Eqs. (8) and (9). Since $x_i^*(t)$ and $\lambda_i^*(t)$ are determined by $x(t)$ and $U(t)$ through Eqs. (1), (2), (8) and (9), Eqs. (3) and (6) can be regarded as an equation defined as

$$F(U(t), x(t), t) := \begin{bmatrix} H_u^T(x_0^*(t), \lambda_1^*(t), u_0^*(t), \mu_0^*(t), p_0^*(t)) \\ C(x_0^*(t), u_0^*(t), p_0^*(t)) \\ \vdots \\ H_u^T(x_{N-1}^*(t), \lambda_N^*(t), u_{N-1}^*(t), \mu_{N-1}^*(t), p_{N-1}^*(t)) \\ C(x_{N-1}^*(t), u_{N-1}^*(t), p_{N-1}^*(t)) \end{bmatrix} = 0.$$

The equation also depends on time t through $p_i^*(t)$ and $\Delta\tau$. If the equation is solved with respect to $U(t)$ for the measured $x(t)$ at each time t, then the control input $u(t) = P_0(U(t))$ is determined.

3 Continuation/GMRES Method

Instead of solving $F(U, x, t) = 0$ itself at each time with such an iterative method as the Newton method, we find the derivative of U with respect to time such that $F(U, x, t) = 0$ is stabilized. Namely, we determine $\dot{U}$ so that

$$\dot{F}(U, x, t) = -\zeta F(U, x, t), \quad (10)$$

where ζ is a positive real number. Then, if F_U is nonsingular, we obtain a differential equation for $U(t)$ as

$$\dot{U} = F_U^{-1}(-\zeta F - F_x \dot{x} - F_t), \quad (11)$$

which is integrated in real time to determine the control input. If $U(0)$ is chosen so that $F(U(0), x(0), 0) = 0$, then the solution $U(t)$ is traced by integrating Eq. (11), which is a kind of the continuation method [5].

From the computational point of view, the differential equation (11) involves expensive operations, i.e., Jacobians F_U, F_x and F_t and a linear equation associated with F_U^{-1}. In order to reduce the computational cost in the Jacobians and the linear equation, we employ two devices, i.e., forward difference approximation for products of Jacobians and vectors, and the GMRES method [3] for the linear equation. First, we approximate the products of the Jacobians and some $W \in \mathbf{R}^{mN}$, $w \in \mathbf{R}^n$ and $\omega \in \mathbf{R}$ with the forward difference as follows.

$$F_U(U, x, t)W + F_x(U, x, t)w + F_t(U, x, t)\omega \simeq D_h F(U, x, t : W, w, \omega) := \frac{F(U + hW, x + hw, t + h\omega) - F(U, x, t)}{h},$$

where h is a positive real number. Then Eq. (10) is approximated by

$$D_h F(U, x, t : \dot{U}, \dot{x}, 1) = -\zeta F(U, x, t),$$

which is equivalent to

$$D_h F(U, x + h\dot{x}, t + h : \dot{U}, 0, 0) = b(U, x, \dot{x}, t), \qquad (12)$$

where

$$b(U, x, \dot{x}, t) := -\zeta F(U, x, t) - D_h F(U, x, t : 0, \dot{x}, 1).$$

It should be noted that the forward difference approximation is different from finite difference approximation of the Jacobians themselves. The forward difference approximation of the products of the Jacobians and vectors can be calculated with only one additional evaluation of the function, which requires notably less computation than approximation of the Jacobians themselves. Because of the forward difference approximation, even a sparse Jacobian is not necessary.

Since Eq. (12) approximates a linear equation with respect to $\dot{U}$, we apply GMRES to Eq. (12) as follows.

Algorithm 1 (`FDGMRES`)
$[\dot{U}, \rho] :=$ `FDGMRES`$(F, U, x, \dot{x}, t, h, k_{max})$

1. $r_0 := b(U, x, \dot{x}, t),\ v_1 := r_0/\|r_0\|,\ \rho := \|r_0\|,\ \beta := \rho,\ k := 0.$
2. While $k < k_{max}$, do
   (a) $k := k + 1$
   (b) $v_{k+1} := D_h F(U, x + h\dot{x}, t + h : v_k, 0, 0)$
   for $j = 1, \ldots, k$
      i. $h_{jk} := v_{k+1}^T v_j$
      ii. $v_{k+1} := v_{k+1} - h_{jk} v_j$
   (c) $h_{k+1,k} := \|v_{k+1}\|$
   (d) $v_{k+1} := v_{k+1}/\|v_{k+1}\|$
   (e) For $e_1 = [1\ 0 \ldots 0]^T \in \mathbf{R}^{k+1}$ and $H_k = (h_{ij}) \in \mathbf{R}^{(k+1)\times k}$ ($h_{ij} = 0$ for $i > j+1$), Minimize $\|\beta e_1 - H_k y^k\|$ to determine $y^k \in \mathbf{R}^k$.
   (f) $\rho := \|\beta e_1 - H_k y^k\|$.
3. $\dot{U} := V_k y^k$, where $V_k = [v_1 \ldots v_k] \in \mathbf{R}^{mN\times k}$.

GMRES is a kind of the Krylov subspace methods for such a linear equation as $Ax = b$ with a nonsymmetric matrix A. GMRES at kth iteration minimizes the residual $\rho := \|b - Ax\|$ with $x \in x_0 + \mathcal{K}_k$, where x_0 is the initial guess and $\mathcal{K}_k$ denotes the Krylov subspace defined by $\mathcal{K}_k := \mathrm{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$ with $r_0 := b - Ax_0$. GMRES also generates an orthonormal basis $\{v_j\}_{j=1}^k$ for $\mathcal{K}_k$ successively. Minimization in step 2e is executed efficiently through use of Givens rotations. In principle, GMRES reduces the residual monotonically and converges to the solution within the same number of iterations as the dimension of the equation. However, an important advantage of GMRES for a large linear equation is that a specified error tolerance, e.g., $\rho \le \eta\|b\|$ ($\eta > 0$), can be achieved with much less iterations.

In `FDGMRES`, the product of a matrix and a vector, $F_U(U, x, t)v_j$, is replaced with its forward difference approximation, $D_h F(U, x + h\dot{x}, t + h : v_k, 0, 0)$. It is clear that the orthonormal basis $\{v_i\}_{i=1}^k$, k vectors in $\mathbf{R}^{mN}$, must be stored during `FDGMRES`, which may require a huge amount of data storage for a large problem. Furthermore, many iterations may be impossible from the viewpoint of execution time in real-time implementation. Therefore, k_{max} should be chosen as small as possible. The state equation (1) and the costate equation (8) are evaluated over the horizon $(3 + k_{max})$ times in `FDGMRES`.

With $\dot{U}$ obtained approximately through use of `FDGMRES`, $U(t)$ is updated by integrating $\dot{U}$ in real time. The continuation/GMRES method for real-time algorithm of nonlinear receding horizon control is summarized as follows.

Algorithm 2 (`C/GMRES`)

1. Let the horizon $T(t)$ a smooth function such that $T(0) = 0$ and $T(t) \to T_f$ $(t \to \infty)$. Let $t = 0$, measure the initial state $x(0)$ and let $x_i^*(0) = x(0)$, $\lambda_i^*(0) = \varphi_x^T(x(0))$ $(i = 0, \ldots, N)$. Find $u(0)$ and $\mu(0)$ analytically or numerically such that
$$\left\| \begin{bmatrix} H_u^T(x(0), \varphi_x^T(x(0)), u(0), \mu(0), p(0)) \\ C(x(0), u(0), p(0)) \end{bmatrix} \right\| \le \frac{\delta}{\sqrt{N}}$$
for some positive δ. Let $u_i^*(0) = u(0)$ and $\mu_i^*(0) = \mu(0)$ $(i = 0, \ldots, N-1)$, which gives the initial condition $U(0)$ such that $\|F(U(0), x(0), 0)\| \le \delta$.
2. For $t' \in [t, t + \Delta t)$, let $U(t') = U(t)$.
3. At time $t + \Delta t$, measure the state $x(t + \Delta t)$ and let $\Delta x = x(t + \Delta t) - x(t)$, compute $\dot{U}(t)$ by $[\dot{U}(t), \rho(t)] =$ `FDGMRES`$(F, U(t), x(t), \Delta x/\Delta t, t, h, k_{max})$. Let $U(t + \Delta t) = U(t) + \dot{U}(t)\Delta t$. The control input $u(t + \Delta t)$ is given by $u(t + \Delta t) = P_0(U(t + \Delta t))$.
4. Let $t = t + \Delta t$, and go back to Step 2.

In `C/GMRES`, explicit solution of $[H_u\ C^T]^T = 0$ is not necessary, since the control input itself is a quantity to be determined numerically. Such higher order derivatives as H_{uu} and H_{ux} are also not necessary, because the linear equation for control update is solved by GMRES with the forward difference approximation. It should be noted that the iterative method is used only to solve the linear equation with respect to $\dot{U}$, and, through use of its solution, the solution of the nonlinear equation, $F(U, x, t) = 0$, is traced without any other iterative methods.

4 Error Analysis

Let X and Ω be subsets of $\mathbf{R}^n$ and $\mathbf{R}^{mN}$, respectively. We assume the following.

A1) $||f(x, P_0(U), p(t))||$, $||F_U^{-1}(U, x, t)||$, $||F_x(U, x, t)||$ and $||F_t(U, x, t)||$ are bounded by a positive constant M for any $(U, x, t) \in \Omega \times X \times [0, \infty)$.

A2) F_U, F_x and F_t are Lipschitz continuous with respect to (U, x, t) on $\Omega \times X \times [0, \infty)$ with a Lipschitz constant α.

We also assume $X' := \{x : \mathcal{B}(x, hM) \subset X\}$ and $\Omega' := \{U : \mathcal{B}(U, h) \subset \Omega\}$ are nonempty.

Since A1) implies F_U is nonsingular, $\mathcal{K}_{mN} = \text{span}\{r_0, F_U r_0, \ldots, (F_U)^{mN-1} r_0\}$ is spanned by mN orthonormal vectors. At kth iteration, FDGMRES tries to generate an orthonormal basis $\{v_j\}_{j=1}^{k}$ that approximates the orthonormal basis of $\mathcal{K}_{mN}$, and its approximation error is $O(h)$ because $||D_h F(U, x + h\dot{x}, t + h : v_j, 0, 0) - F_U(U, x, t) v_j||$ is $O(h)$. Therefore, if h is chosen sufficiently small, FDGMRES generates orthonormal vectors $\{v_j\}_{j=1}^{k_{max}}$ for any $k_{max} \leq mN$. We assume $\{v_j\}_{j=1}^{k_{max}}$ are orthonormal hereafter.

Let $\{\bar{v}_j\}_{j=1}^{mN}$ be an orthonormal basis for $\mathbf{R}^{mN}$ such that $\bar{v}_j = v_j$ for $i = 1, \ldots, k_{max}$, and define a matrix $B \in \mathbf{R}^{mN \times mN}$ by

$$B\bar{v}_j = D_h F(U, x + h\dot{x}, t + h : \bar{v}_j, 0, 0) \quad (j = 1, \ldots, mN).$$

Then, FDGMRES is nothing but GMRES for a linear equation $B\dot{U} = b$ with k_{max} iterations. Therefore, the residual $\rho = ||b - B\dot{U}||$ reduces monotonically with respect to k_{max} and converges to zero for some $k_{max} \leq mN$. Particularly, for any $\eta > 0$, FDGMRES guarantees $\rho \leq \eta ||b||$ if k_{max} is sufficiently large. However, we need some analysis concerning the accuracy of the resultant value of $\dot{F}$, since B is not the exact Jacobian of F. We analyze a residual r in Eq. (10) defined by

$$\begin{aligned} r(U, x, \dot{U}, \dot{x}, t) &:= F_U(U, x, t)\dot{U} + F_x(U, x, t)\dot{x} \\ &\quad + F_t(U, x, t) + \zeta F(U, x, t) \end{aligned}$$

which indicates to what extent $\dot{F}$ differs from the desirable value given in Eq. (10). Only main results are presented without proofs because of limitation of space.

Theorem 1 *Let $\eta \in (0, 1)$, $(U, x, t) \in \Omega' \times X' \times [0, \infty)$, $\dot{x} = f(x, P_0(U), p(t))$, and $\dot{U}$ be obtained by* FDGMRES, *then there exists $\bar{h} > 0$ such that if $h \in (0, \bar{h}]$ and* FDGMRES *terminates with $\rho \leq \eta ||b||$ then*

$$||r|| \leq \hat{\eta}\zeta ||F|| + \hat{M},$$

where

$$\begin{aligned} \hat{\eta} &:= \eta + 2\hat{h}(1 + \eta), \\ \hat{h} &:= \alpha h \sqrt{k_{max}} \left(\frac{3}{2} + M \right) M, \\ \hat{M} &:= \hat{\eta}(1 + M)M + (1 + \hat{\eta} + 4\hat{h})\hat{\beta}, \\ \hat{\beta} &:= \frac{\alpha h}{2}(M^2 + 2M + 1). \end{aligned}$$

Theorem 1 gives a bound for r when $\dot{U}$ is calculated by FDGMRES. Finally, an error bound is analyzed for C/GMRES in which FDGMRES is used. We denote $F(U(l\Delta t), x(l\Delta t), l\Delta t)$ by F_l for short.

Lemma 1 *Let the system be controlled by* C/GMRES. *There exists $\bar{h} > 0$ such that if $h \in (0, \bar{h}]$,* FDGMRES *always terminates with $\rho \leq \eta ||b||$ for a some $\eta \in (0, 1)$, and $(U(l\Delta t), x(l\Delta t)) \in \Omega' \times X'$ then*

$$||F_{l+1}|| \leq \delta_1 ||F_l||^2 + \delta_2 ||F_l|| + \delta_3,$$

where

$$\begin{aligned} \delta_1 &:= a_1 \zeta^2 \Delta t^2, \quad a_1 := \frac{\alpha \eta'^2}{2}, \quad \eta' := 2(1 + \eta)M, \\ \delta_2 &:= |1 - \zeta\Delta t| + \hat{\eta}\zeta\Delta t + a_2 \zeta \Delta t^2, \\ a_2 &:= \frac{\alpha \eta'}{2}(1 + 2\beta' + 2\alpha M), \\ \beta' &:= M(2(1 + \eta)M^2 + (6 + 2\eta)\hat{\beta}), \\ \delta_3 &:= \hat{M}\Delta t + a_3 \Delta t^2, \\ a_3 &:= \frac{\alpha}{2}(M + \beta' + 1)^2. \end{aligned}$$

Theorem 2 *Let the system be controlled by* C/GMRES *with sufficiently small h and Δt such that Lemma 1 holds and*

$$\hat{\eta} + a_2 \Delta t \leq 1 - 2\sqrt{a_1 \Delta t(\hat{M} + a_3 \Delta t)} \tag{13}$$

for a some $\eta \in (0, 1)$. Furthermore, let ζ and δ in C/GMRES *be chosen so that*

$$0 < \zeta \Delta t \leq \bar{\zeta} \Delta t, \tag{14}$$

$$\bar{\zeta}\Delta t := \frac{2}{1 + \hat{\eta} + a_2 \Delta t + 2\sqrt{a_1 \Delta t(\hat{M} + a_3 \Delta t)}}, \tag{15}$$

$$\delta := \frac{1 - \delta_2 - \sqrt{(1 - \delta_2)^2 - 4\delta_1 \delta_3}}{2\delta_1}. \tag{16}$$

If FDGMRES *always terminates with $\rho \leq \eta ||b||$, and $(U(t), x(t)) \in \Omega' \times X'$ for all $t \in [0, \infty)$, then*

$$||F_l|| \leq \delta,$$

for all $l \in \mathbf{Z}_+$. The minimum value of δ is attained when $\zeta = 1/\Delta t$.

Note that $1/\Delta t \leq \bar{\zeta} < 2/\Delta t$ and $\delta > 0$ hold from Eqs. (13) and (15). Therefore, it is always possible to choose $\zeta = 1/\Delta t$, and ζ cannot exceed $2/\Delta t$.

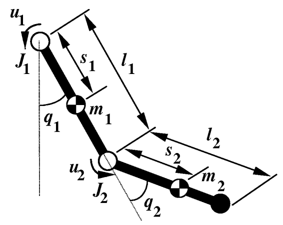

Figure 1: Two-link arm.

5 Numerical Example

In order to evaluate the computational time of the proposed algorithm `C/GMRES`, we employ a two-link arm in a vertical plane (Fig. 1).

The state equation of the two-link arm is given by

$$\frac{d}{dt}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ M^{-1}(q)(u - V(q,\dot{q}) - G(q)) \end{bmatrix}$$

where $q = [q_1 \; q_2]^T$ is a vector of joint angles, $u = [u_1 \; u_2]^T$ is a vector of joint torques, $M(q) \in \mathbf{R}^{2\times 2}$ is an inertia matrix, $V(q,\dot{q}) \in \mathbf{R}^2$ is a vector of centrifugal and Coriolis terms, and $G(q) \in \mathbf{R}^2$ is a vector of gravity terms. The physical parameters are given as: $l_1 = 0.3$ m, $s_1 = 0.15$ m, $m_1 = 0.2$ kg, $J_1 = 6.0\times10^{-3}$ kg·m^2, $l_2 = 0.3$ m, $s_2 = 0.257$ m, $m_2 = 0.7$ kg, and $J_2 = 5.1\times10^{-2}$ kg·m^2, respectively.

Although the present system is rather simple as a robotic system, it is highly nonlinear and its real-time optimal control is extremely difficult. We would like to control this system with receding horizon control whose performance index is given by

$$\begin{aligned} J &= \frac{1}{2}(x(t+T) - x_f)^T S_f (x(t+T) - x_f) \\ &\quad + \frac{1}{2}\int_t^{t+T} ((x - x_f)^T Q (x - x_f) + u^T R u) dt'. \end{aligned}$$

The state vector of the present system is $x = [q^T \; \dot{q}^T]^T \in \mathbf{R}^4$, $x_f \in \mathbf{R}^4$ denotes the objective state, and S_f, Q and R are weighting matrices.

The simulation program is generated in C through use of an automatic code generation system, `AutoGenU` [6]. Simulation is performed on a personal computer (CPU: Pentium II, 300 MHz) with $k_{max} = 2$ and $\zeta = 1/\Delta t$. The control input is updated by `C/GMRES`, and the state equation is integrated with the Adams method on the t axis for simulation. The objective state in this case is $x_f = [\pi \; 0 \; 0 \; 0]^T$. The length of the horizon is chosen so that $T(0) = 0$ and $T(t) \to T_f$ as $t \to \infty$, namely, $T(t) := T_f(1 - e^{-\alpha t})$ with $T_f = 0.2$ s and $\alpha = 1.5$.

Figure 2 compares the computational time required by `C/GMRES` for simulation of control process of 1.5 s. The weighting matrices are chosen as $S_f = \text{diag}[4, 2, 0.001, 0.001]$, $Q = \text{diag}[40, 20, 0.01, 0.01]$ and $R = I$, respectively. Those weights are chosen so that the conventional algorithm can also generates an accurate simulation result. It is apparent that a smaller integration step, Δt, on the t axis requires a larger computational time, and, at the same time, the computational time is proportional to the number of grids on the horizon. The computational time is less than 1.5 s below the dashed line in the figure, and therefore, real-time control is possible with the simulation conditions in that region.

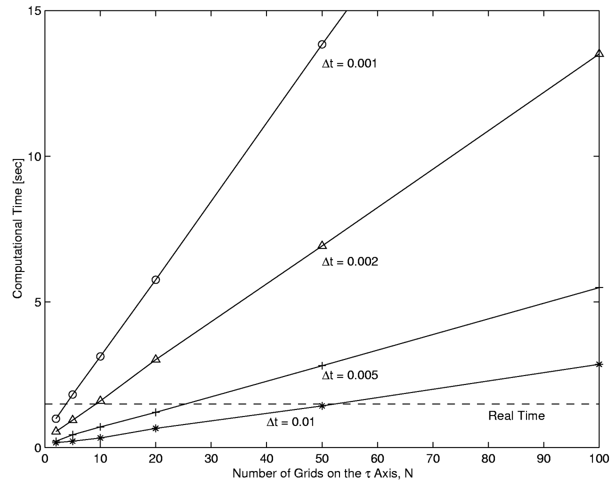

Figure 2: Computational time for simulation of 1.5 s with $k_{max} = 2$.

On the other hand, the conventional algorithm [7, 8] which updates the costate based on the Euler-Lagrange equations often fails and requires a large number of grids on the horizon, e.g., $N = 100$. The conventional algorithm takes 63 s for $\Delta t = 0.001$ s, 32 s for $\Delta t = 0.002$ s, 13 s for $\Delta t = 0.005$ s and 6.5 s for $\Delta t = 0.01$ s, which indicates that real-time control is not possible for the present system with the conventional algorithm. The proposed algorithm achieves a notable improvement concerning the computational time.

Finally, as an example of problems which are difficult for the conventional algorithms, we treat a swing-up control of the two-link arm with only one input, u_1, with a magnitude constraint, $|u_1| \le u_{1max}$. Since the system is underactuated and constrained, it cannot be controlled by standard control techniques for robotic manipulators. The inequality constraint on u_1 is converted into an equality constraint: $C(u_1, u_3) := (u_1^2 + u_3^2 - u_{1max}^2)/2 = 0$, where u_3 denotes a dummy input. The maximal input magnitude is chosen as $u_{1max} = 4$. The weighting matrices are chosen as $S_f = \text{diag}[6, 2, 0.001, 0.001]$, $Q = \text{diag}[6, 2, 0.004, 0.004]$, respectively.

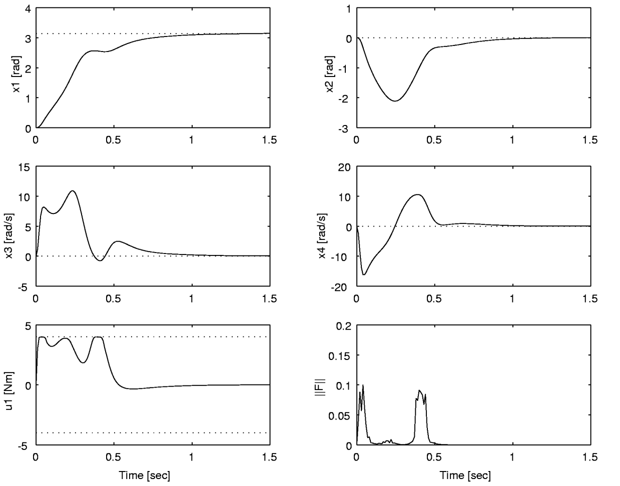

Figure 3: Time histories of swing-up control with only one constrained actuator.

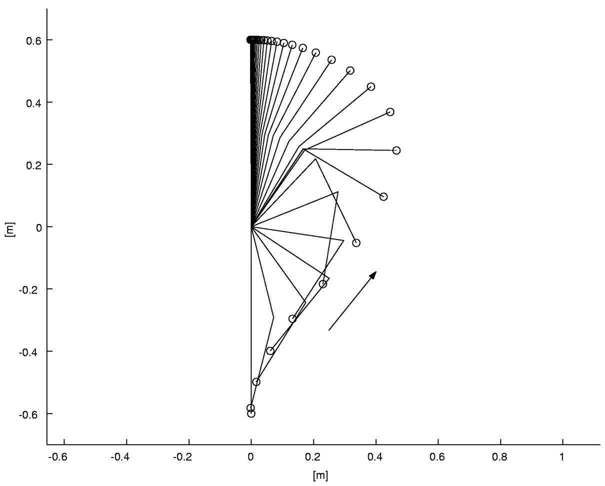

Figure 4: Swing-up motion of the two-link arm with only one constrained actuator.

The computational time by C/GMRES is less than 1.5 s for simulation of 1.5 s with $\Delta t = 0.002$ s, $N = 5$ and $k_{max} = 4$, which indicates that real-time control is possible with a sampling interval of 0.002 s. The simulation result with C/GMRES is shown in Figs. 3 and 4. The two-link arm is successfully swung up by only one constrained actuator. In contrast, the conventional algorithm fails even if $\Delta t = 0.001$ s in the present problem. The proposed algorithm is much faster and numerically more robust than the conventional algorithm.

6 Conclusions

This paper has proposed a real-time algorithm for nonlinear receding horizon control by combining the continuation method with GMRES. The error analysis has shown that with a sufficiently small step in the forward difference approximation, GMRES with the forward difference approximation, named `FDGMRES`, yields an accurate solution. It has also been shown that the error in the optimality condition for the discretized problem is bounded with the control input updated by the proposed algorithm named `C/GMRES`.

The proposed algorithm has been demonstrated in a numerical example of a two-link arm whose dynamics is highly nonlinear. Numerical study has shown that `C/GMRES` is much faster and numerically more robust than the conventional algorithm based on the differential equation of the unknown costate. The highly nonlinear two-link arm can be controlled with the proposed algorithm in real time under such conditions as underactuation and magnitude constraint.

Acknowledgements

This work was partially supported by the Research Fund of the Kurata Memorial Science and Technology Promotion Foundation and a Grant-in-Aid for Scientific Research from Ministry of Education, Japan (No. 11750178).

References

[1] E. S. Meadows and J. B. Rawlings, "Model predictive control," in *Nonlinear Process Control*, M. A. Henson and D. E. Seborg, Eds., chapter 5, pp. 233–310. Prentice Hall, NJ, 1997.

[2] P. Lu, "Optimal predictive control of continuous nonlinear systems," *International Journal of Control*, vol. 62, no. 3, pp. 633–649, 1995.

[3] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, vol. 16 of *Frontiers in Applied Mathematics*, SIAM, Philadelphia, PA, 1995.

[4] U. M. Ascher, R. M. M. Mattheji, and R. D. Russel, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, SIAM, Philadelphia, PA, 1995.

[5] S. L. Richter and R. A. DeCarlo, "Continuation methods: Theory and applications," *IEEE Transactions on Automatic Control*, vol. AC-28, no. 6, pp. 660–665, 1983.

[6] T. Ohtsuka, *http://www-newton.mech.eng.osaka-u.ac.jp/˜ohtsuka/code/index.htm*, 2000.

[7] T. Ohtsuka and H. A. Fujii, "Real-time optimization algorithm for nonlinear receding-horizon control," *Automatica*, vol. 33, no. 6, pp. 1147–1154, 1997.

[8] T. Ohtsuka, "Time-variant receding-horizon control of nonlinear systems," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 174–176, 1998.