Available online at www.sciencedirect.com
SCIENCE DIRECT®
Automatica 40 (2004) 563–574
automatica
www.elsevier.com/locate/automatica

A continuation/GMRES method for fast computation of nonlinear receding horizon control☆

Toshiyuki Ohtsuka*

Department of Computer-Controlled Mechanical Systems, Graduate School of Engineering, Osaka University, 2-1 Yamadaoka, Suita, Osaka 565-0871, Japan

Received 17 December 2001; received in revised form 21 October 2003; accepted 13 November 2003

Abstract

In this paper, a fast numerical algorithm for nonlinear receding horizon control is proposed. The control input is updated by a differential equation to trace the solution of an associated state-dependent two-point boundary-value problem. A linear equation involved in the differential equation is solved by the generalized minimum residual 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 some conditions. The proposed algorithm is applied to a two-link arm whose dynamics is highly nonlinear. Simulation results show that the proposed algorithm is faster than the conventional algorithms.

Keywords: Receding horizon control; Nonlinear control; Real-time optimization; Continuation method; GMRES

1. Introduction

Receding horizon control (model predictive control) is a potential control technique for nonlinear systems. In receding horizon control, an open-loop optimal control problem, which leads to a two-point boundary-value problem (TPBVP), is solved in real time. There are two issues concerning nonlinear receding horizon control: how to choose problem settings to guarantee closed-loop stability, and how to compute the control input in real time. From the theoretical point of view, several stability conditions have been obtained (see, e.g, Mayne, Rawlings, Rao, & Scokaert, 2000; Fontes, 2001). Those conditions concerning stability give guidelines for problem settings and selection of a performance index, although those conditions are applicable only in rather restricted situations.

The main concern of this paper is numerical algorithms for nonlinear receding horizon control. As the capabilities of digital computers advance, receding horizon control is becoming successfully applied in the chemical industry (Meadows & Rawlings, 1997; Qin & Badgwell, 2000), where the sampling period is sufficiently large, e.g., several tens of seconds or longer, and an iterative optimization method can be executed to solve the optimization problem within the sampling period. However, an iterative optimization method is computationally expensive and is not suitable for mechanical systems controlled with a sampling period in the order of milliseconds. Efficient numerical algorithms should be explored to broaden the applications of receding horizon control.

Although a control law of nonlinear receding horizon control can be obtained explicitly through the use of the Taylor expansion (Lu, 1995), the length of the horizon and the form of the performance index are restricted in such an algorithm. In general, the direct optimization of a control law requires a solution of the Hamilton–Jacobi–Bellman equation (HJBE). There have been numerous studies on solving the HJBE. Since the HJBE is a partial differential equation of the optimal cost function, its solution must be given over some region in the state space. The structure of the solution is usually assumed for the HJBE, e.g., in the form of a power series (Al'Brekht, 1961), interpolation (Kreisselmeier & Birkhölzer, 1994), a neural network (Goh, 1993), or an expansion with basis functions (Beard, Saridis,

☆ This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Tor Arne Johansen under the direction of Editor Frank Allgöwer.
* Tel.: +81-6-6789-7329; fax: +81-6-6879-7247.
E-mail address: ohtsuka@mech.eng.osaka-u.ac.jp (T. Ohtsuka).

0005-1098/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2003.11.005

& Wen, 1998; Alamir, 2001). There is also an approximate solution technique that does not assume the structure of the solution in the framework of genetic programming (Imae & Takahashi, 1999). However, those methods cannot avoid the explosive growth of data storage or number of terms for high-dimensional systems and are difficult to implement for a system whose dimension is higher than two or three.

If the solution of the HJBE is not available and one has to solve the nonlinear receding horizon control problem in real time, it is natural to exploit a particular structure of the problem to realize an efficient numerical algorithm. For example, one can apply iterative optimization methods utilizing the specific structure of the Hessian (Biegler, 2000; Tousain & Bosgra, 2000; Diehl et al., 2002). Alternatively, if an unconstrained optimal control law is known, one can shift on-line computational burden to off-line computational burden for the constrained case, by perturbing the unconstrained control law to guarantee feasibility (Cannon, Kouvaritakis, Lee, & Brooms, 2001). However, iterative optimization methods are still computationally expensive, and an optimal control law is not established for a nonlinear system even in the unconstrained case.

Another feature of the nonlinear receding horizon control problem is that its solution varies smoothly with respect to time under a certain condition. Therefore, it is possible to trace the solution without iterative optimization methods, which is nothing but the continuation method (homotopy method) (Richter & DeCarlo, 1983; Allgower & Georg, 1990). In fact, the differential equation of the unknown costate is obtained as a variant of the Euler–Lagrange equations and can be utilized as a real-time algorithm (Ohtsuka & Fujii, 1997; Ohtsuka, 1998). However, that algorithm involves a complicated Riccati differential equation for solving a linear TPBVP associated with the time-dependent variation in the optimal trajectory.

In this paper, a real-time algorithm of nonlinear receding horizon control, in which the continuation method is combined with a fast algorithm for linear equations instead of the Riccati differential equation, is proposed. The formulation and implementation of the proposed algorithm are substantially different from those of the previous algorithm, although both algorithms are based on the continuation method. First, the open-loop optimal control problem is discretized, resulting in a nonlinear algebraic equation for the discretized sequence of control input. This equation determines the optimal control sequence as an implicit function of the current state and time. Then, a differential equation for updating the control sequence is obtained through the use of the continuation method. That is, the derivative of the control sequence with respect to time is determined according to the corresponding derivative of the state with respect to time, and it can be integrated to determine the control input in real time without iterative optimization methods. Since that differential equation involves a large linear equation, we need a fast algorithm to solve it. We employ the generalized minimum residual (GMRES) method (Kelley, 1995; Barrett et al., 1994), which is a Krylov subspace method, to solve the linear equation. From the viewpoint of real-time control, the proposed method has the favorable feature that the computational time is almost constant for updating the control input, while the computational time varies in an iterative optimization method in general.

Although the proposed algorithm is expressed as a differential equation obtained from the continuation method, it must be discretized with respect to time for implementation, and the algorithm may fail because of discretization error. Therefore, 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 and numerical error. Simulation results show that real-time control is possible for a highly nonlinear system, using the proposed algorithm. It is also shown by simulation that the result of the error analysis gives a useful guideline for choosing the free parameters in the algorithm.

This paper is organized as follows: in Section 2 the nonlinear receding horizon control problem is summarized. The discretized problem and the nonlinear equation for the control sequence are presented in Section 3. The real-time algorithm called C/GMRES is proposed in Section 4, in which the continuation method is combined with GMRES, and the boundedness of its error is analyzed in Section 5 with technical details presented in Appendices. The numerical example is presented in Section 6, and some conclusions are given in Section 7.

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 $\mathscr{B}(x_0,\varepsilon) := \{x : \|x - x_0\| \leqslant \varepsilon\}$. All functions are assumed to have continuous partial derivatives of any required order.

2. Receding horizon control problem

In this section, the nonlinear receding horizon control problem is briefly summarized. We consider a general nonlinear system governed by the state equation

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

where $x(t) \in \mathbb{R}^n$ denotes the state vector, $u(t) \in \mathbb{R}^{m_u}$ the input vector, and $p(t) \in \mathbb{R}^{m_p}$ 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:

$$J = \varphi(x^u(t+T; t, x(t)), p(t+T)) + \int_t^{t+T} L(x^u(t'; t, x(t)), u(t'), p(t'))\,\mathrm{d}t',$$

where $x^u(t'; t, x(t))$ $(t \leqslant t' \leqslant t+T)$ denotes the state trajectory by the input function u starting from $x(t)$ at time t. That is, the actual state $x(t)$ is used as the initial state for the

optimal control problem over the horizon ($t \leqslant t' \leqslant t+T$). Equality constraints are also imposed over the horizon in general as

$$C(x^u(t';t,x(t)),u(t'),p(t'))=0,$$

where C is an m_c-dimensional vector-valued function. In the case of an inequality constraint, it is necessary to introduce a penalty function method or some heuristic modification of the problem, as is presented in the numerical example. The optimal control u_{opt} is determined as a function over the horizon and also depends on t and $x(t)$ as $u_{\mathrm{opt}}(t';t,x(t))$ $(t \leqslant t' \leqslant t+T)$. However, the actual input to the systems is given only by the value of u_{opt} at time t, that is, $u(t)=u_{\mathrm{opt}}(t;t,x(t))$, which is a state feedback control law.

The receding horizon control problem is essentially a family of finite horizon optimal control problems along a fictitious time τ as follows:

$$\text{Minimize:}\quad J=\varphi(x^*(T,t),p(t+T))+\int_0^T L(x^*(\tau,t),u^*(\tau,t),p(t+\tau))\,\mathrm{d}\tau$$

$$\text{Subject to:}\begin{cases} x^*_\tau(\tau,t)=f(x^*(\tau,t),u^*(\tau,t),p(t+\tau)),\\ x^*(0,t)=x(t),\\ C(x^*(\tau,t),u^*(\tau,t),p(t+\tau))=0. \end{cases}$$

The new state vector $x^*(\tau,t)$ denotes the trajectory along the τ axis starting from $x(t)$ at $\tau=0$. The optimal control input $u^*(\tau,t)$ is determined on the τ axis as the solution of the finite horizon optimal control problem for each t, and the actual control input is given by $u(t)=u^*(0,t)$. The horizon T is a function of time, $T=T(t)$, in general, as is explained later.

3. Discretized problem

We divide the horizon into N steps and discretize the optimal control problem on the τ-axis with the forward difference as follows:

$$x^*_{i+1}(t)=x^*_i(t)+f(x^*_i(t),u^*_i(t),p^*_i(t))\Delta\tau(t), \tag{1}$$

$$x^*_0(t)=x(t), \tag{2}$$

$$C(x^*_i(t),u^*_i(t),p^*_i(t))=0, \tag{3}$$

$$J=\varphi(x^*_N(t),p^*_N(t))+\sum_{i=0}^{N-1} L(x^*_i(t),u^*_i(t),p^*_i(t))\Delta\tau(t), \tag{4}$$

where $\Delta\tau(t) := T(t)/N$, $x^*_i(t)$ corresponds to $x^*(i\Delta\tau,t)$, and $p^*_i(t)$ is given by $p(t+i\Delta\tau)$. Since the horizon T depends on time t in general, so does $\Delta\tau$. Note that only the problem on the τ axis is discretized, and the dependence on t remains continuous at this stage of problem formulation. That is, the discretized problem is solved at each continuous time t, although the time t is also discretized eventually in the actual implementation. 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

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

where $\lambda\in\mathbb{R}^n$ denotes the costate, and $\mu\in\mathbb{R}^{m_c}$ denotes the Lagrange multiplier associated with the equality constraint. The first-order necessary conditions for the sequences of the 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

$$H_u(x^*_i(t),\lambda^*_{i+1}(t),u^*_i(t),\mu^*_i(t),p^*_i(t))=0, \tag{5}$$

$$\lambda^*_i(t)=\lambda^*_{i+1}(t)+H_x^{\mathrm{T}}(x^*_i(t),\lambda^*_{i+1}(t),u^*_i(t),\mu^*_i(t),p^*_i(t))\Delta\tau(t), \tag{6}$$

$$\lambda^*_N(t)=\varphi_x^{\mathrm{T}}(x^*_N(t),p^*_N(t)). \tag{7}$$

The sequences of the optimal control $\{u^*_i(t)\}_{i=0}^{N-1}$ and the multiplier $\{\mu^*_i(t)\}_{i=0}^{N-1}$ must satisfy (1)–(3) and (5)–(7), which define a TPBVP for the discretized optimal control problem. It should be noted that the TPBVP for the discretized problem is identical to a finite difference approximation of the TPBVP for the original continuous-time problem. Therefore, the solution of the discretized problem converges to the solution of the continuous-time problem as $N\to\infty$ under mild conditions (Ascher, Mattheji, & Russel, 1995).

We define a vector of the input and multipliers as

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

where $m := m_u+m_c$. We also define a projection $P_0 : \mathbb{R}^{mN}\to\mathbb{R}^{m_u}$ as $P_0(U) := u^*_0$. For a given $U(t)$ and $x(t)$, $\{x^*_i(t)\}_{i=0}^{N}$ is calculated recursively by (1) and (2), and then, $\{\lambda^*_i(t)\}_{i=0}^{N}$ is also calculated recursively from $i=N$ to $i=0$, by (6) and (7). Since $x^*_i(t)$ and $\lambda^*_i(t)$ are determined by $x(t)$ and $U(t)$ through (1), (2), (6) and (7), Eqs. (3) and (5) can be regarded as one equation defined as

$$F(U(t),x(t),t) := \begin{bmatrix} H_u^{\mathrm{T}}(x^*_0(t),\lambda^*_1(t),u^*_0(t),\mu^*_0(t),p^*_0(t))\\ C(x^*_0(t),u^*_0(t),p^*_0(t))\\ \vdots\\ H_u^{\mathrm{T}}(x^*_{N-1}(t),\lambda^*_N(t),u^*_{N-1}(t),\mu^*_{N-1}(t),p^*_{N-1}(t))\\ C(x^*_{N-1}(t),u^*_{N-1}(t),p^*_{N-1}(t)) \end{bmatrix}=0. \tag{8}$$

This equation also depends on time t through $p_i^*(t)$ and $\Delta\tau(t)$. 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.

Other high-order discretization schemes can also be employed to obtain the equation at the expense of simplicity and computational burden. In particular, generalization of (1) and (6) to any explicit scheme is straightforward. In the case of implicit schemes, however, recursive calculation of $\{x_i^*(t)\}_{i=0}^N$ and $\{\lambda_i^*(t)\}_{i=0}^N$ involves the solution of nonlinear equations at each step i and is computationally demanding. One can avoid such recursive calculation by regarding all of the state sequence and costate sequence as independent variables rather than as functions of U. In that case, the number of unknown variables and the size of the equation increase according to the dimension of the state. On the other hand, in the present formulation of (8), the number of unknown variables mN does not depend on the dimension of the state, which is preferable for high-dimensional systems.

4. Continuation/GMRES method

4.1. Continuation method

Instead of solving $F(U,x,t) = 0$ itself at each time with such an iterative method as Newton's method, we find the derivative of U with respect to time such that $F(U(t),x(t),t) = 0$ is satisfied identically. Namely, we choose $U(0)$ so that $F(U(0),x(0),0) = 0$ and determine $\dot{U}$ so that

$$\dot{F}(U,x,t) = A_s F(U,x,t), \tag{9}$$

where A_s is a stable matrix introduced to stabilize $F = 0$. Then, if F_U is nonsingular, we obtain a differential equation for $U(t)$ as

$$\dot{U} = F_U^{-1}(A_s F - F_x \dot{x} - F_t), \tag{10}$$

which can also be regarded as a linear algebraic equation, with a coefficient matrix F_U, to determine $\dot{U}$ for given $U, x, \dot{x}$ and t. We can update the solution $U(t)$ of $F(U(t),x(t),t)=0$ without iterative optimization methods by integrating (10) in real time. The present approach is a kind of continuation method (Richter & DeCarlo, 1983; Allgower & Georg, 1990) in the sense that the solution curve $U(t)$ is traced by integrating a differential equation.

In practical implementation, $U(0)$ that satisfies $F(U(0),x(0),0) = 0$ must be found through some numerical method, and $\dot{x}$ in (10) must be approximated by finite difference, as is explained in Section 4.3. Moreover, numerical integration of $\dot{U}$ imposes a certain condition on matrix A_s to maintain boundedness of the error F, as analyzed in Section 5.

4.2. Forward difference GMRES method

From the computational point of view, the differential eqution (10) still involves expensive operations, i.e., Jacobians F_U, F_x and F_t and a linear equation associated with F_U^{-1}. It should be noted that, in the present definition of F in (8), the Jacobian F_U is dense because $x_i^*(t)$ and $\lambda_i^*(t)$ are functions of $U(t)$, and therefore sparse finite difference is not applicable. 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 (Kelley, 1995; Barrett et al., 1994) for the linear equation. First, we approximate the products of the Jacobians and some $W \in \mathbb{R}^{mN}$, $w \in \mathbb{R}^n$ and $\omega \in \mathbb{R}$ with the forward difference as follows:

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

where h is a positive real number. Then (9) is approximated by

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

which is equivalent to

$$D_h F(U, x + h\dot{x}, t + h : \dot{U}, 0, 0) = b(U,x,\dot{x},t), \tag{11}$$

where

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

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 an additional evaluation of the function, which requires notably less computational burden than approximation of the Jacobians themselves.

Since (11) approximates a linear equation with respect to $\dot{U}$, we apply GMRES to (11), which is called FDGMRES in Kelley (1995). Here, we modify and generalize FDGMRES slightly to include a certain initial guess $\hat{\dot{U}}$ as follows.

Algorithm 1. $\dot{U} := \texttt{FDGMRES}(U,x,\dot{x},t,\hat{\dot{U}},h,k_{\max})$

(1) $\hat{r} := b(U,x,\dot{x},t) - D_h F(U, x + h\dot{x}, t + h : \hat{\dot{U}}, 0, 0)$, $v_1 := \hat{r}/\|\hat{r}\|$, $\rho := \|\hat{r}\|$, $\beta := \rho$, $k := 0$.
(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) $h_{jk} := v_{k+1}^{\mathrm{T}} v_j$
    (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* $e_1=[1\,0\ldots0]^{\mathrm{T}} \in \mathbb{R}^{k+1}$ *and* $H_k=(h_{ij}) \in \mathbb{R}^{(k+1)\times k}$ $(h_{ij}=0$ *for* $i>j+1)$, *Minimize* $\|\beta e_1 - H_k y^k\|$ *to determine* $y^k \in \mathbb{R}^k$.
(f) $\rho := \|\beta e_1 - H_k y^k\|$.

(3) $\dot{U} := \hat{\dot{U}} + V_k y^k$, *where* $V_k = [v_1 \ldots v_k] \in \mathbb{R}^{mN\times k}$.

GMRES is a kind of Krylov subspace method for such a linear equation as $Ax = b$ with a nonsymmetric matrix A. GMRES at the kth iteration minimizes the residual $\rho := \|b - Ax\|$ with $x \in x_0 + \mathscr{K}_k$, where x_0 is the initial guess and $\mathscr{K}_k$ denotes the Krylov subspace defined by $\mathscr{K}_k := \mathrm{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$ with $r_0 := b - Ax_0$. GMRES also successively generates an orthonormal basis $\{v_j\}_{j=1}^k$ for $\mathscr{K}_k$. Minimization in step 2e is executed efficiently through the 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., $\rho \leqslant \eta\|\hat{r}\|$ $(0 < \eta < 1)$, can be achieved with much fewer 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, $D_hF(U, x + h\dot{x}, t + h : v_k, 0, 0)$. It is clear that the orthonormal basis $\{v_i\}_{i=1}^k$, k vectors in $\mathbb{R}^{mN}$, must be stored during execution of FDGMRES, which may require a huge amount of data storage for a large problem. Moreover, many iterations may be impossible from the viewpoint of execution time in real-time implementation. The state equation (1) and the costate equation (6) are evaluated over the horizon $3 + k_{\max}$ times in FDGMRES. Therefore, the iteration number $k_{\max}$ should be chosen as small as possible. Fortunately, in the present real-time application, the solution $\dot{U}$ at the previous sampling time is often a good initial guess for FDGMRES and a small $k_{\max}$ often suffices to obtain an accurate solution. If a large number of iterations are necessary with a limited amount of data storage, it is a common technique to restart FDGMRES from the current iterate by resetting the orthonormal basis.

It is also well known that appropriate preconditioning of the equation improves the convergence of such iterative methods as GMRES. The basic idea of preconditioning is to transform a linear equation $Ax = b$ to $(M_1^{-1}AM_2^{-1})(M_2x) = M_1^{-1}b$ with M_1 and M_2 chosen such that M_1M_2 approximates A. The transformed equation for M_2x has an improved convergence property, and x is obtained from M_2x. The matrices M_1 and M_2 are called the left and right preconditioners, respectively. There is a trade-off between additional computational cost for preconditioning and an improved convergence property. Effective preconditioning is problem dependent and various preconditioners have been proposed (Barrett et al., 1994; Kelley, 1995). In the case of nonlinear receding horizon control, scaling of the input vector u is effective if magnitudes are substantially different between the input channels. Extensive study on preconditioning for nonlinear receding horizon control is beyond the scope of this paper.

4.3. Combination of continuation and GMRES

With $\dot{U}$ obtained approximately through the use of FDGMRES, $U(t)$ is updated by integrating $\dot{U}$ in real time. For a sampling period Δt, a variable at time $l\Delta t$ is denoted with the suffix (l) for short hereafter. The continuation/GMRES method for nonlinear receding horizon control is summarized as follows.

Algorithm 2 (C/GMRES). (1) *Setting* $t := 0$ *and* $l := 0$, *measure the initial state* $x_{(0)} := x(0)$ *and find* $U_{(0)}$ *analytically or numerically such that* $\|F(U_{(0)}, x_{(0)}, 0)\| \leqslant \delta$ *for some positive* δ.

(2) *For* $t' \in [t, t + \Delta t)$, *set* $u(t') := P_0(U_{(l)})$.

(3) *At time* $t+\Delta t$, *measure the state* $x_{(l+1)} := x(t+\Delta t)$, *set* $\Delta x_{(l)} := x_{(l+1)} - x_{(l)}$, *and compute* $\dot{U}_{(l)}$ *by*

$$\dot{U}_{(l)} := \texttt{FDGMRES}(U_{(l)}, x_{(l)}, \Delta x_{(l)}/\Delta t, t, \hat{\dot{U}}_{(l)}, h, k_{\max}),$$

where the initial guess $\hat{\dot{U}}_{(l)}$ *is appropriately chosen*; *e.g.*, $\hat{\dot{U}}_{(l)} := 0$ or $\hat{\dot{U}}_{(l)} := \dot{U}_{(l-1)}$ *with* $\hat{\dot{U}}_{(-1)} := 0$. *Set* $U_{(l+1)} := U_{(l)} + \Delta U_{(l)}$ *with* $\Delta U_{(l)} := \dot{U}_{(l)}\Delta t$.

(4) *Set* $t := t + \Delta t$, $l := l + 1$, *and go back to Step* 2.

It should be noted that the iterative method is used only to solve the linear equation (10) with respect to $\dot{U}$, and, through the use of its solution, the solution of the nonlinear equation, $F(U,x,t) = 0$, is traced without any line search or Newton iteration. C/GMRES solves the linear equation (10) only once at each sampling time and, therefore, requires much less computational burden than such iterative methods as Newton's method which solves a linear equation several times to determine search directions. Moreover, C/GMRES involves no line search, which is also a significant difference from standard optimization methods.

A simple method for initializing $U_{(0)}$ is to choose the horizon $T(t)$ as a smooth function such that $T(0) = 0$ and $T(t) \to T_f$ $(t \to \infty)$. Then, $u_i^*(0) = u(0)$ and $\mu_i^*(0) = \mu(0)$ $(i=0,\ldots,N-1)$, $x_i^*(0)=x(0)$ and $\lambda_i^*(0)=\varphi_x^{\mathrm{T}}(x(0))$ $(i=0,\ldots,N)$, and the initialization is reduced to finding $u(0)$ and $\mu(0)$ such that

$$\left\| \begin{bmatrix} H_u^{\mathrm{T}}(x(0), \varphi_x^{\mathrm{T}}(x(0)), u(0), \mu(0), p(0)) \\ C(x(0), u(0), p(0)) \end{bmatrix} \right\| \leqslant \frac{\delta}{\sqrt{N}},$$

which needs much less computational burden than the original Eq. (8). If C/GMRES needs to be extended for nonlinear receding horizon control with a terminal constraint, the zero horizon length is not preferable because the initial state $x(0)$ already has to satisfy the terminal condition. However, it would be possible to achieve the terminal constraint asymptotically by introducing a time-dependent penalty function or appropriate continuation in the terminal constraint.

Alternatively, if sufficient computational time is available before starting control, one can choose a nonzero initial horizon length by initializing $U(0)$ with Newton's method or a gradient method.

In C/GMRES, since the control input itself is a quantity to be determined numerically, explicit solution of $H_u = 0$ is not necessary, which is advantageous for nonaffine systems. Such higher order derivatives as H_{uu}, H_{ux} and H_{xx} contained in F_U are also not necessary, because the linear equation for control update is solved by GMRES with the forward difference approximation. Moreover, the Hessian of the Hamiltonian H with respect to u and μ,

$$\begin{bmatrix} H_{uu} & C_u^{\mathrm{T}} \\ C_u & 0 \end{bmatrix},$$

may be singular as long as the entire Jacobian F_U is nonsingular, which is in contrast to some of the existing numerical methods for optimal control problems (Miele, Mangiavacchi, & Aggarwal, 1974; Ohtsuka & Fujii, 1994). In other words, when the TPBVP is regarded as a differential–algebraic equation (DAE) for the differential variables x and λ and the algebraic variables u and μ, its index can be higher than one as long as F_U is nonsingular. This is possible because $x_i^*(t)$ and $\lambda_i^*(t)$ are functions of $U(t)$, and therefore, the off-diagonal blocks of F_U are nonzero in the present formulation, as was mentioned previously. For example, even if the constraint $C = 0$ does not contain the input u and $C_u = 0$, the off-diagonal blocks of F_U contain such terms as $C_x(x_i^*)(\partial x_i^*/\partial u_j^*)$ $(i > j)$, where $\partial x_i^*/\partial u_j^*$ denotes the sensitivity of x_i^* with respect to the past input u_j^* and is determined by (1). If F_U is singular, $\dot{U}$ does not exist or is not unique, which corresponds to vanishment or bifurcation of the solution curve $U(t)$, respectively, and C/GMRES fails.

5. Error analysis

5.1. Assumptions

The entire algorithm C/GMRES involves several approximations such as the forward difference approximation of the product of the Jacobian and a vector, truncation in GMRES, and numerical integration of $\dot{U}$. Therefore, stability of $F = 0$ imposed by the original condition (9) is not necessarily guaranteed in the practical implementation. This section presents an analytical result of the error $\|F\|$ in the practical implementation, taking those approximations into account.

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

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

Assumption 2. 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 K.

Assumption 3. $\bar{X} := \{x : \mathscr{B}(x, hM) \subset X\}$ and $\bar{\Omega} := \{U : \mathscr{B}(U,h) \subset \Omega\}$ are nonempty.

Assumption 4. For all $t \in [0,\infty)$ and all $l \in \mathbb{Z}_{\geqslant 0}$, $x(t) \in \bar{X}$ and $U_{(l)} \in \bar{\Omega}$.

Assumption 5. $h > 0$ is sufficiently small so that $\hat{h} := Kh\sqrt{k_{\max}}(M + 3/2)M \leqslant 1/2$.

Assumption 6. For some $\eta \in (0,1)$, FDGMRES always terminates with $\rho \leqslant \eta\|\hat{r}\|$.

Even if there are disturbances or uncertainties in the actual system, the result of the error analysis in this section is valid as long as the above assumptions hold. Although Assumption 4 requires that the state and control input in the closed-loop system remain on certain sets, an asymptotically stable equilibrium point is not assumed to exist. That is, the present error analysis is applicable to such general problems as time-varying systems, trajectory tracking, stabilization of a periodic orbit, and practical stabilization. In these general problems, the error $\|F\|$ is not necessarily expected to converge to zero because of the persistent approximation error at every sampling time. Therefore, we guarantee the boundedness of the error, which is fundamental for a practical control system and contrasts with a standard error analysis to guarantee convergence of a numerical method.

It should also be noted that the boundedness and Lipschitz continuity of the Jacobians in Assumptions 1 and 2 are related to the open-loop characteristics of the system. For example, if the system has an unstable equilibrium point, the sensitivity of the trajectory $\{x_i^*\}_{i=0}^N$ with respect to the input sequence $\{u_i^*\}_{i=0}^{N-1}$ and the initial state $x_0^* = x$ can be large near the unstable equilibrium point, and, consequently, the norm bound M and the Lipschitz constant K for the Jacobians can also be large. However, the sensitivity of the trajectory can be reduced by choosing a shorter horizon length. The norm bound M and the Lipschitz constant K specify a possible choice of the sampling period Δt and the matrix A_s to guarantee the boundedness of the error, as is shown in the main result.

5.2. Main result

In this subsection, we summarize the main result of the error analysis. It is natural to use $\dot{U}$ at the previous sampling time as the initial guess for FDGMRES; we call this a warm start. The condition for the boundedness of the error $\|F\|$ is given as follows.

Theorem 7. *Let the system be controlled by C/GMRES with* $\hat{\dot{U}}_{(l)} = \dot{U}_{(l-1)}$, *and let Assumptions* 1–6 *hold. Let* η *be*

sufficiently small and let $A_s \neq 0$, h *and* Δt *be chosen so that*

$$\|I + A_s \Delta t\| + \eta'' \|A_s\| \Delta t \leqslant 1, \tag{12}$$

$$\hat{\eta} M^2 < 1 - \kappa \bar{a} M < 1, \tag{13}$$

where η'', $\hat{\eta}$, κ *and* $\bar{a}$ *are constants given in Appendix* A. *Then there exist positive constants* δ_F, δ_r *and* δ_U *such that, for* $l \in \mathbb{Z}_{\geqslant 0}$, $\|F_{(l)}\| \leqslant \delta_F$, $\|r_{(l)}\| \leqslant \delta_r$ *and* $\|\dot{U}_{(l)}\| + M + 1 \leqslant \delta_U$ *imply* $\|F_{(l+1)}\| \leqslant \delta_F$, $\|r_{(l+1)}\| \leqslant \delta_r$ *and* $\|\dot{U}_{(l+1)}\| + M + 1 \leqslant \delta_U$.

The proof of Theorem 7 is presented in Appendix B. Equation (12) gives a condition for the free parameters A_s, h and Δt, while (13) imposes an additional condition that η is not only less than 1 but also sufficiently small, which cannot be guaranteed a priori. For the particular case of $A_s = -\zeta I$ ($\zeta > 0$), we can obtain a more transparent condition for ζ, h and Δt as follows.

Corollary 8. *Let the system be controlled by C/GMRES with* $\hat{\dot{U}}_{(l)} = \dot{U}_{(l-1)}$, *and let Assumptions* 1–6 *hold. Let* η, h *and* Δt *be sufficiently small such that* (13) *and* $\eta'' \leqslant 1$ *hold. Moreover, let* $A_s = -\zeta I$ ($\zeta > 0$) *be chosen so that*

$$0 < \zeta \Delta t \leqslant \frac{2}{1 + \eta''}. \tag{14}$$

Then, there exist positive constants δ_F, δ_r *and* δ_U *such that, for* $l \in \mathbb{Z}_{\geqslant 0}$, $\|F_{(l)}\| \leqslant \delta_F$, $\|r_{(l)}\| \leqslant \delta_r$ *and* $\|\dot{U}_{(l)}\| + M + 1 \leqslant \delta_U$ *imply* $\|F_{(l+1)}\| \leqslant \delta_F$, $\|r_{(l+1)}\| \leqslant \delta_r$ *and* $\|\dot{U}_{(l+1)}\| + M + 1 \leqslant \delta_U$.

Corollary 8 is a direct consequence of Theorem 7, and its proof is omitted. Note that $1 \leqslant 2/(1 + \eta'') < 2$ if $\eta'' \leqslant 1$. Then, it is always possible to choose $\zeta \Delta t = 1$, and the algorithm may fail if $\zeta \Delta t$ exceeds 2. It should also be noted that C/GMRES with $A_s = -\zeta I$ corresponds to a one step relaxed Newton method with a relaxation parameter $\zeta \Delta t$ and tangential predictor for solution of $F(U, x, t) = 0$.

The case with $\hat{\dot{U}}_{(l)} = 0$, which we call a cold start, can also be analyzed by a similar approach as the warm start. A similar condition as Theorem 7 is obtained for the cold start, which is omitted here.

6. Numerical example

6.1. Two-link arm

In order to evaluate the computational time and numerical error of the proposed algorithm C/GMRES, we consider a two-link arm in a vertical plane (Fig. 1). The physical parameters are given as $l_1 = 0.3$ m, $s_1 = 0.15$ m, $m_1 = 0.2$ kg, $J_1 = 6.0 \times 10^{-3}$ kg m^2, $l_2 = 0.3$ m, $s_2 = 0.257$ m, $m_2 = 0.7$ kg, and $J_2 = 5.1 \times 10^{-2}$ kg m^2.

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

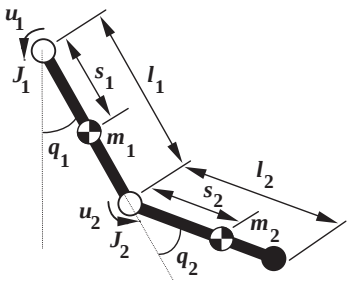

Fig. 1. Two-link arm.

$$\varphi := \tfrac{1}{2} (x - x_f)^{\mathrm{T}} S_f (x - x_f),$$

$$L := \tfrac{1}{2} ((x - x_f)^{\mathrm{T}} Q (x - x_f) + u^{\mathrm{T}} R u).$$

The state vector of the present system is $x = [q^{\mathrm{T}} \ \dot{q}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^4$, $x_f \in \mathbb{R}^4$ denotes the objective state, and S_f, Q and R are weighting matrices. Since the Hessian of the Hamiltonian, H_{uu}, equals R, the obtained stationary solution is a local minimum solution if R is positive definite and there is no conjugate point.

Simulation is performed on a personal computer (CPU: Pentium II, 300 MHz) with $k_{\max} = 2$. The control input is updated by C/GMRES with $\hat{\dot{U}}_{(l)} = \dot{U}_{(l-1)}$ (warm start), and the state equation is integrated with the Adams method on the t-axis. The objective state in this case is $x_f = [\pi \ 0 \ 0 \ 0]^{\mathrm{T}}$. The length of the horizon is chosen so that $T(0) = 0$ and $T(t) \to T_f$ as $t \to \infty$, namely, $T(t) := T_f(1 - \mathrm{e}^{-\alpha t})$ with $T_f = 0.2$ s and $\alpha = 1.5$. The simulation program in C language is generated by an automatic code generation system called AutoGenU (Ohtsuka, 2000). AutoGenU is a Mathematica program for generating a simulation program for nonlinear receding horizon control.

6.2. Computational time and accuracy

First, the proposed algorithm C/GMRES is compared with a conventional algorithm (Ohtsuka, 1998) based on the backward sweep method, and two general-purpose optimization codes based on the limited-memory BFGS method (L-BFGS) (Liu & Nocedal, 1989; Nocedal, 1990) and on the truncated Newton method (Nash, 1984a, b). For a control problem with m inputs with the horizon divided into N grid points, the number of optimized variables is mN in all the methods except the backward sweep method in which the costate λ_0^* is determined instead of the control input sequence U. In the L-BFGS and the truncated Newton method, U is directly optimized at each time with the initial guess given by the optimal value at the previous time step.

For the simulation of the control process of 1.5 s, Table 1 lists the average computational times per update and

Table 1
Average computational time per update (ms) (upper line) and the average error per grid point $\|F\|/N$ (lower line)

N	C/GMRES	Backward sweep	L-BFGS	Truncated Newton
5	0.9 3.12×10^{-3}	Failure	1.8 2.74×10^{-3}	1.4 1.16×10^{-3}
10	1.6 3.04×10^{-3}	Failure	4.6 3.19×10^{-3}	3.1 2.35×10^{-3}
20	3.3 2.82×10^{-3}	Failure	17.7 2.72×10^{-3}	26.2 2.71×10^{-3}
50	7.9 2.32×10^{-3}	Failure	48.9 8.31×10^{-3}	63.9 8.92×10^{-3}
100	15.4 1.86×10^{-3}	34.8 (0.1113)	96.3 1.84×10^{-2}	211.2 2.20×10^{-2}

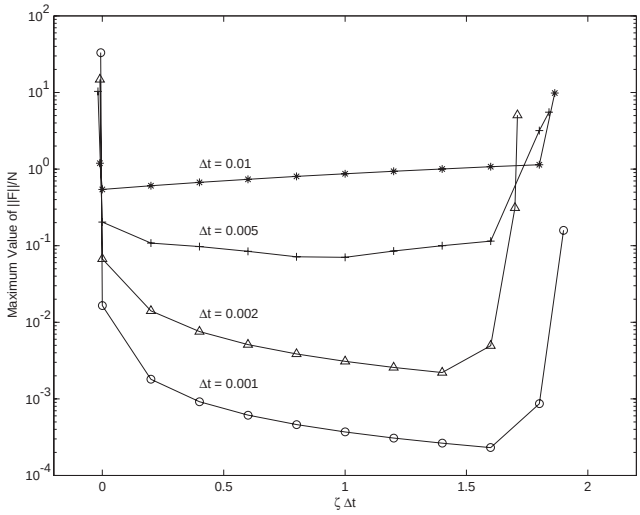

Fig. 2. The maximal error per grid point $\|F\|/N$ versus the value of $\zeta\Delta t$.

the average error per grid point $\|F\|/N$. Only the error of the backward sweep method is defined differently from others as $\|\lambda_N^* - \varphi_x^{\mathrm{T}}(x_N^*, p_N^*)\|$. The weighting matrices are chosen as $S_f = \mathrm{diag}[4, 2, 0.001, 0.001]$, $Q = \mathrm{diag}[40, 20, 0.01, 0.01]$ and $R = I$. The same time-dependent horizon length $T(t) = T_f(1 - \mathrm{e}^{-\alpha t})$ is used for all the algorithms. Tuning parameters in the L-BFGS and the truncated Newton method are chosen so that the errors are of an order comparable to those in C/GMRES. That is, all the numerical algorithms except the backward sweep method are compared at the same performance level. The computational time per update represents the average time required to update U at each time step in the simulation and gives a rough estimate of a possible sampling period for implementation in the real world.

It is apparent from the table that C/GMRES is much faster than other methods, and the computational time of C/GMRES increases linearly with respect to the number of grid points, N, while the computational times of L-BFGS and the truncated Newton increase rapidly as the number of grid points increases. This observation is expected because C/GMRES updates U with the least amount of computation without iterative line searches. It should be noted, although not observed in the table, that the computational time for update in C/GMRES is almost constant, which is preferable for real-time control, while the computational time for update in an iterative optimization method varies in general. Since the backward sweep method updates the costate, which is similar to the shooting method, it is sensitive to numerical error in comparison with other methods of updating the control sequence and fails in the present highly nonlinear problem unless the number of grid points is sufficiently large. Although the backward sweep method does not fail and is faster than the L-BFGS and the truncated Newton method for $N = 100$, it is not faster than C/GMRES because it requires a Riccati differential equation to be solved over the horizon, which involves the complicated derivative H_{xx}.

Next, the actual behavior of the error in the simulation is compared with the stability condition obtained by the error analysis in the previous section. The maximum value of the normalized error $\|F\|/N$ is plotted in Fig. 2 for the parameter $\zeta\Delta t$. The horizon is divided into $N = 20$ steps. It is confirmed from Fig. 2 that the lower limit of $\zeta\Delta t$ is zero, and the upper limit is between 1 and 2, as is predicted by the error analysis. Therefore, it is concluded that the stability condition in the error analysis gives a useful guideline for selecting $\zeta\Delta t$. Since the optimal value of $\zeta\Delta t$ depends on the control problem and computational conditions, $\zeta\Delta t = 1$ would be a simple and practical selection.

6.3. Constrained problem

Finally, as an example of problems with constraints, we treat a swing-up control of the two-link arm with only one constrained actuator. Since the system is underactuated, it cannot be controlled by standard control techniques for robotic manipulators. Only the control input u_1 is used, and its magnitude is constrained as $|u_1| \leqslant u_{1\max}$. The maximum magnitude is set to $u_{1\max} = 4$. The inequality constraint is converted into an equality constraint by introducing a dummy input u_3 as follows:

$$C(u_1, u_3) := (u_1^2 + u_3^2 - u_{1\max}^2)/2 = 0.$$

It should be noted that the dummy input u_3 is squared to constrain the sign of $u_1^2 - u_{1\max}^2$ to be nonpositive. Since the sign of the dummy input u_3 does not affect the optimality, the solution curve $U(t)$ can bifurcate and C/GMRES cannot determine the update of u_3 when $u_3 = 0$. To avoid the singularity at $u_3 = 0$, a small dummy penalty is also added to the performance index. That is, the performance index is

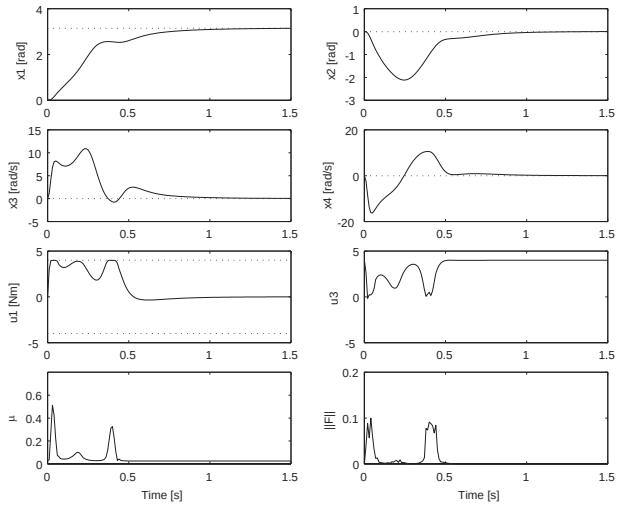

Fig. 3. Time histories of swing-up control with only one constrained actuator.

modified so that positive value of u_3 is preferable, as

$$L := \tfrac{1}{2}\left((x - x_f)^{\mathrm{T}} Q (x - x_f) + r_1 u_1^2 - r_3 u_3\right),$$

where r_3 is a small positive constant. Then, u_3 must be non-zero owing to $H_{u_3} = \mu u_3 - r_3 = 0$. Although the present treatment of the inequality constraint is heuristic, it is useful in practical applications because it makes the present algorithm applicable to a wider class of problems without significant modification and avoids possible numerical difficulties in penalty function methods. In addition to u_1 and u_3, the Lagrange multiplier μ is also included in the optimized variable U in this problem. The parameters are chosen as $S_f = \mathrm{diag}[6, 2, 0.001, 0.001]$, $Q = \mathrm{diag}[6, 2, 0.004, 0.004]$, $r_1 = 1$, $r_3 = 0.1$, $\Delta t = 0.002$ s, $N = 5$, $k_{\max} = 4$, and $\zeta = 1/\Delta t$. The computational time by C/GMRES is less than 1 s for simulation of 1.5 s, which indicates that real-time control is possible with a sampling period of 2 ms. The simulation result with C/GMRES is shown in Fig. 3. The two-link arm is successfully swung up by only one constrained actuator.

7. Conclusions

In this paper, a numerical algorithm for nonlinear receding horizon control has been proposed. The original continuous-time problem was first discretized over the horizon, and a TPBVP was obtained to determine the sequence of control input for the discretized problem. Since the control input is chosen as the unknown variable to be determined and is updated by the numerical method, the optimality condition for the control input, $H_u = 0$, need not be solved explicitly.

The derivative of the control input sequence with respect to time was determined so that the solution of the TPBVP was traced with respect to time, which is a kind of continuation method and results in a linear equation involving Jacobians. The linear equation was solved by GMRES, one of the Krylov subspace methods, with the forward difference approximation of products of the Jacobians and vectors, which requires less computational cost than direct approximation of the Jacobians themselves.

It has been shown, taking errors in GMRES and the forward difference approximation into account, that the error in the entire algorithm, named C/GMRES, is bounded under a certain condition. The stability condition in the error analysis gives a useful guideline for selecting free parameters in C/GMRES.

The proposed algorithm has been demonstrated for a numerical example of a two-link arm whose dynamics is highly nonlinear. Numerical study has shown that C/GMRES is much faster than the conventional algorithms. The highly nonlinear two-link arm can be controlled with the proposed algorithm in real time under such conditions as underactuation and constraints.

Acknowledgements

This work is partially supported by the Research Fund of the Kurata Memorial Science and Technology Promotion Foundation, a Grant-in-Aid for Scientific Research (Nos. 11750178, 15760319) from the Ministry of Education, Culture, Sports, Science and Technology, Japan, and Grants for Receivers of Doctorates from the Foundation for C&C Promotion.

Appendix A. Constants in error analysis

$$\hat{h} := Kh\sqrt{k_{\max}}(M + 3/2)M,$$

$$\hat{\eta} := \eta + 2\hat{h}(1 + \eta),$$

$$M' := (M^2 + 1)(M + 1),$$

$$\kappa := (1 + \hat{\eta} + 4\hat{h})h/\Delta t,$$

$$a_{1r} := \frac{\hat{\eta}(1 + M^2)\|A_{\mathrm{s}}\|\Delta t + \kappa(1 - \|I + A_{\mathrm{s}}\Delta t\|)}{(1 + \kappa - \hat{\eta}M^2)\Delta t},$$

$$a_{2r} := \frac{\hat{\eta}MM'}{1 + \kappa - \hat{\eta}M^2},$$

$$a_{1U} := M(\|A_{\mathrm{s}}\| + a_{1r}),$$

$$a_{2U} := Ma_{2r} + M',$$

$$\bar{a} := Ka_{2U}\Delta t + \sqrt{2Ka_{2r}\Delta t + (Ka_{2U}\Delta t)^2},$$

$$\eta'' := \frac{\hat{\eta}(1+M^2)+(1+\hat{\eta}+\kappa)\bar{a}M}{1-\hat{\eta}M^2-\kappa\bar{a}M},$$

$$a_{1F} := K(a_{1U}\Delta t)^2/2,$$

$$a_{2F} := \|I + A_s\Delta t\| + a_{1r}\Delta t + Ka_{1U}a_{2U}\Delta t^2,$$

$$a_{3F} := a_{2r}\Delta t + K(a_{2U}\Delta t)^2/2.$$

Appendix B. Proof of the main result

B.1. Analysis of FDGMRES

First, we analyze the errors in FDGMRES, which is necessary to analyze the entire algorithm C/GMRES. For the orthonormal basis $\{v_j\}_{j=1}^{k_{\max}}$ generated by FDGMRES, let $\{\bar{v}_j\}_{j=1}^{mN}$ be an orthonormal basis for $\mathbb{R}^{mN}$ such that $\bar{v}_j = v_j$ for $j = 1,\ldots,k_{\max}$, and define a matrix $B \in \mathbb{R}^{mN\times mN}$ such that

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

Then, FDGMRES is nothing but GMRES for the linear equation $B\dot{U} = b$ with $k_{\max}$ iterations. Therefore, the residual $\rho = \|\hat{r} - B(\dot{U} - \hat{\dot{U}})\|$ decreases monotonically during the iteration. In particular, for any $\eta > 0$, FDGMRES guarantees $\rho \leqslant \eta\|\hat{r}\|$ if $k_{\max}$ is sufficiently large. However, we need some analysis of the accuracy of the resultant value of $\dot{F}$, since B is not the exact Jacobian of F. We analyze the residual r in (9) defined by

$$r(U,x,\dot{U},\dot{x},t) := F_U(U,x,t)\dot{U} + F_x(U,x,t)\dot{x} + F_t(U,x,t) - A_sF(U,x,t), \tag{B.1}$$

which indicates to what extent $\dot{F}$ differs from the desirable value given in (9). Through the use of the following lemmas, we obtain bounds for r and $\hat{r}$ in FDGMRES.

Lemma B.1. *Let Assumptions* 1–4 *hold. Let* $\dot{U}$ *be obtained by FDGMRES with* $k_{\max}$ *iterations, then*

$$\|(B - F_U(U,x,t))(\dot{U} - \hat{\dot{U}})\| \leqslant \hat{h}\|F_U(U,x,t)(\dot{U} - \hat{\dot{U}})\|.$$

Lemma B.2. *Let Assumptions* 1–4 *hold. Then*

$$\|F_U(U,x,t)\hat{\dot{U}} + F_x(U,x,t)\dot{x} + F_t(U,x,t) - D_hF(U,x,t:\hat{\dot{U}},\dot{x},1)\| \leqslant \frac{Kh}{2}(\|\hat{\dot{U}}\| + M + 1)^2.$$

The proofs of Lemmas B.1 and B.2 are similar to that of Proposition 6.2.1 in (Kelley, 1995) and are omitted. Through the use of Lemmas B.1 and B.2, we can bound the residual r defined by (B.1) and the initial residual $\hat{r}$, defined in FDGMRES, corresponding to the initial guess $\hat{\dot{U}}$, as follows.

Lemma B.3. *Let Assumptions* 1–6 *hold. Let* $\dot{U}$ *be obtained by FDGMRES, then*

$$\|r\| \leqslant \hat{\eta}\|\hat{r}\| + (1+4\hat{h})\frac{Kh}{2}(\|\hat{\dot{U}}\| + M + 1)^2,$$

$$\|\hat{r}\| \leqslant \|A_s\|\|F(U,x,t)\| + M(\|\hat{\dot{U}}\| + M + 1) + \frac{Kh}{2}(\|\hat{\dot{U}}\| + M + 1)^2,$$

where $\hat{\eta}$ *is a constant given in Appendix* A.

Proof. Let the arguments of F, F_U, F_x and F_t be (U,x,t) throughout the proof. From the definitions of r, $b(U,x,\dot{x},t)$ and $\hat{r}$, the residual r can be rewritten as

$$r = -\hat{r} + B(\dot{U} - \hat{\dot{U}}) - (B - F_U)(\dot{U} - \hat{\dot{U}}) + F_U\hat{\dot{U}} + F_x\dot{x} + F_t - D_hF(U,x,t:\hat{\dot{U}},\dot{x},1). \tag{B.2}$$

From the definitions of $\hat{r}$ and B and Assumption 6, we have $\|\hat{r} - B(\dot{U} - \hat{\dot{U}})\| \leqslant \eta\|\hat{r}\|$. Then, through the use of Lemmas B.1 and B.2, (B.2) implies

$$-\|\hat{r}\| + \|F_U(\dot{U} - \hat{\dot{U}})\| - \frac{Kh}{2}(\|\hat{\dot{U}}\| + M + 1)^2 \leqslant \|r\| \leqslant \eta\|\hat{r}\| + \hat{h}\|F_U(\dot{U} - \hat{\dot{U}})\| + \frac{Kh}{2}(\|\hat{\dot{U}}\| + M + 1)^2. \tag{B.3}$$

From (B.3) and Assumption 5,

$$\|F_U(\dot{U} - \hat{\dot{U}})\| \leqslant 2(1+\eta)\|\hat{r}\| + 2Kh(\|\hat{\dot{U}}\| + M + 1)^2. \tag{B.4}$$

Then, (B.3) and (B.4) imply the first inequality.

To prove the second inequality, we rewrite $\hat{r}$ as

$$\hat{r} = A_sF - (F_U\hat{\dot{U}} + F_x\dot{x} + F_t) + F_U\hat{\dot{U}} + F_x\dot{x} + F_t - D_hF(U,x,t:\hat{\dot{U}},\dot{x},1).$$

Then, Assumption 1 and Lemma B.2 imply the second inequality. □

B.2. Analysis of C/GMRES

Next, we analyze the propagation of the error in C/GMRES through the use of the previous bounds for FDGMRES. The fundamental theorem of calculus is used to express the error in numerical integration of $\dot{U}$.

Lemma B.4. *Let the system be controlled by C/GMRES, and let Assumptions* 1–6 *hold, then the following holds*:

$$\|F_{(l+1)}\| \leqslant \|I + A_s\Delta t\|\|F_{(l)}\| + \|r_{(l)}\|\Delta t + \frac{K\Delta t^2}{2}(\|\dot{U}_{(l)}\| + M + 1)^2,$$

$$\|r_{(l+1)}\| \leqslant \hat{\eta}\|A_s\|\|F_{(l+1)}\| + \hat{\eta}M(\|\hat{\dot{U}}_{(l+1)}\| + M + 1) + (1 + \hat{\eta} + 4\hat{h})\frac{Kh}{2}(\|\hat{\dot{U}}_{(l+1)}\| + M + 1)^2,$$

$$\|\dot{U}_{(l+1)}\| + M + 1 \leqslant M(\|A_s\|\|F_{(l+1)}\| + \|r_{(l+1)}\|) + M',$$

where M' is a constant defined in Appendix A.

Proof. From the fundamental theorem of calculus,

$$F_{(l+1)} = F_{(l)} + \int_0^1 \frac{\mathrm{d}}{\mathrm{d}\xi} F(U_{(l)} + \xi\Delta U_{(l)}, x_{(l)} + \xi\Delta x_{(l)}, t_{(l)} + \xi\Delta t)\,\mathrm{d}\xi.$$

Then, the first inequality is obtained through the use of $F_{U(l)}\Delta U_{(l)} + F_{x(l)}\Delta x_{(l)} + F_{t(l)}\Delta t = A_s F_{(l)}\Delta t + r_{(l)}\Delta t$, $\Delta U_{(l)} = \dot{U}_{(l)}\Delta t$, $\|\Delta x_{(l)}\| \leqslant M\Delta t$ and Assumption 2. The second inequality is a direct consequence of Lemma B.3. The third inequality is obtained readily from

$$\dot{U}_{(l+1)} = F_{U(l+1)}^{-1}(A_s F_{(l+1)} - F_{x(l+1)}(\Delta x_{(l+1)}/\Delta t) - F_{t(l+1)} + r_{(l+1)}). \qquad \square$$

Now we can prove Theorem 7 through the use of Lemma B.4. Our approach is to find positive numbers δ_F, δ_r and δ_U such that the equalities in Lemma B.4 are satisfied with $\|F_{(l+1)}\| = \|F_{(l)}\| = \delta_F$, $\|r_{(l+1)}\| = \|r_{(l)}\| = \delta_r$ and $\|\dot{U}_{(l)}\| + M + 1 = \|\hat{\dot{U}}_{(l+1)}\| + M + 1 = \|\dot{U}_{(l+1)}\| + M + 1 = \delta_U$. Then, since all coefficients in the inequalities are positive, $\|F_{(l)}\| \leqslant \delta_F$, $\|r_{(l)}\| \leqslant \delta_r$ and $\|\dot{U}_{(l)}\| + M + 1 = \|\hat{\dot{U}}_{(l+1)}\| + M + 1 \leqslant \delta_U$ imply $\|F_{(l+1)}\| \leqslant \delta_F$, $\|r_{(l+1)}\| \leqslant \delta_r$ and $\|\dot{U}_{(l+1)}\| + M + 1 \leqslant \delta_U$, as is claimed.

Proof of Theorem 7. Consider the following simultaneous equations for δ_F, δ_r and δ_U:

$$\|I + A_s\Delta t\|\delta_F + \delta_r\Delta t + \frac{K\Delta t^2}{2}\delta_U^2 = \delta_F, \tag{B.5}$$

$$\hat{\eta}\|A_s\|\delta_F + \hat{\eta}M\delta_U + (1 + \hat{\eta} + 4\hat{h})\frac{Kh}{2}\delta_U^2 = \delta_r, \tag{B.6}$$

$$M(\|A_s\|\delta_F + \delta_r) + M' = \delta_U. \tag{B.7}$$

Elimination of δ_U^2 and δ_U in (B.6) using (B.5) and (B.7) implies $\delta_r = a_{1r}\delta_F + a_{2r}$, and its substitution into (B.7) implies $\delta_U = a_{1U}\delta_F + a_{2U}$, where a_{1r}, a_{2r}, a_{1U} and a_{2U} are constants presented in Appendix A. Then (B.5) is rewritten as a quadratic equation for δ_F with two solutions δ_{F+} and δ_{F-}, as

$$\delta_{F\pm} := \frac{1 - a_{2F} \pm \sqrt{D_F}}{2a_{1F}}, \tag{B.8}$$

$$D_F := (1 - a_{2F})^2 - 4a_{1F}a_{3F}, \tag{B.9}$$

where a_{1F}, a_{2F} and a_{3F} are also presented in Appendix A.

It is apparent from (B.8) and (B.9) that $0 < \delta_{F-} \leqslant \delta_{F+}$ if $a_{1F}a_{3F} > 0$ and $1 - a_{2F} \geqslant 2\sqrt{a_{1F}a_{3F}}$, i.e., $a_{2F} + 2\sqrt{a_{1F}a_{3F}} \leqslant 1$. First, both a_{1F} and a_{3F} are positive by their definitions if $1 + \kappa - \hat{\eta}M^2 > 0$. Next, we have

$$a_{2F} + 2\sqrt{a_{1F}a_{3F}} = \|I + A_s\Delta t\| + a_{1r}\Delta t + \bar{a}a_{1U}\Delta t.$$

Since $a_{1r}\Delta t$ and $a_{1U}\Delta t$ have a common denominator, $1+\kappa-\hat{\eta}M^2$, the condition $a_{2F} + 2\sqrt{a_{1F}a_{3F}} \leqslant 1$ can be rearranged if $1 + \kappa - \hat{\eta}M^2 > 0$, as

$$(1 - \hat{\eta}M^2 - \kappa\bar{a}M)\|I + A_s\Delta t\| + (\hat{\eta}(1 + M^2) + (1 + \hat{\eta} + \kappa)\bar{a}M)\|A_s\|\Delta t \leqslant 1 - \hat{\eta}M^2 - \kappa\bar{a}M.$$

Moreover, if $1 - \hat{\eta}M^2 - \kappa\bar{a}M > 0$, the condition is rewritten as (12). Condition (13) guarantees both $1+\kappa-\hat{\eta}M^2 > 0$ and $1 - \hat{\eta}M^2 - \kappa\bar{a}M > 0$. That is, conditions (12) and (13) guarantee $a_{1F}a_{3F} > 0$, $1 - a_{2F} \geqslant 2\sqrt{a_{1F}a_{3F}}$ and, consequently, $0 < \delta_{F-} \leqslant \delta_{F+}$.

Conditions (12) and (13) also guarantee that all coefficients a_{1r}, a_{2r}, a_{1U} and a_{2U} are positive by definition. Therefore, $\delta_{r\pm} := a_{1r}\delta_{F\pm} + a_{2r}$ and $\delta_{U\pm} := a_{1U}\delta_{F\pm} + a_{2U}$ are all positive. Finally, Lemma B.4 and (B.5)–(B.7) guarantee that $(\delta_F, \delta_r, \delta_U)$ chosen as $(\delta_{F+}, \delta_{r+}, \delta_{U+})$ or $(\delta_{F-}, \delta_{r-}, \delta_{U-})$ has the desired property. $\square$

References

Alamir, M. (2001). Solutions of nonlinear optimal and robust control problems via a mixed collocation/DAE's based algorithm. *Automatica*, *37*, 1109–1115.

Al'Brekht, E. G. (1961). On the optimal stabilization of nonlinear systems. *Journal of Applied Mathematics and Mechanics*, *25*(5), 1254–1266.

Allgower, E. L., & Georg, K. (1990). *Numerical continuation methods*. Heiderberg: Springer.

Ascher, U. M., Mattheji, R. M. M., & Russel, R. D. (1995). *Numerical solution of boundary value problems for ordinary differential equations*. Philadelphia, PA: SIAM.

Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., & Van der Vorst, H. (1994). *Templates for the solution of linear systems: Building blocks for iterative methods* (2nd ed.). Philadelphia, PA: SIAM.

Beard, R. W., Saridis, G. N., & Wen, J. T. (1998). Approximate solutions to the time-invariant Hamilton–Jacobi–Bellman equation. *Journal of Optimization Theory and Applications*, *96*(3), 589–626.

Biegler, L. T. (2000). Efficient solution of dynamic optimization and NMPC problems. In F. Allgöwer, & A. Zheng (Eds.), *Nonlinear model predictive control* (pp. 219–243). Basel: Birkhäuser.

Cannon, M., Kouvaritakis, B., Lee, Y. I., & Brooms, A. C. (2001). Efficient non-linear model based predictive control. *International Journal of Control*, *74*(4), 361–372.

Diehl, M., Bock, H. G., Findeisen, R., Schlöder, J. P., Nagy, Z., & Allgöwer, F. (2002). Real-time optimization and nonlinear model predictive control of process governed by differential-algebraic equations. *Journal of Process Control*, *12*(4), 577–585.

Fontes, F. A. C. C. (2001). A general framework to design stabilizing nonlinear model predictive controllers. *Systems and Control Letters*, *42*, 127–143.

Goh, C. J. (1993). On the nonlinear optimal regulator problem. *Automatica*, *29*(3), 751–756.

Imae, J., & Takahashi, J. (1999). A design method for nonlinear H_∞ control systems via Hamilton–Jacobi–Isaacs equations: A genetic programming approach. In *Proceedings of the 38th IEEE conference on decision and control*, Phoenix, AZ (pp. 3782–3783).

Kelley, C. T. (1995). Iterative methods for linear and nonlinear equations. In *Frontiers in applied mathematics*, Vol. 16. Philadelphia, PA: SIAM.

Kreisselmeier, G., & Birkhölzer, T. (1994). Numerical nonlinear regulator design. *IEEE Transactions on Automatic Control*, *39*(1), 33–46.

Liu, D. C., & Nocedal, J. (1989). On the limited memory method for large scale optimization. *Mathematical Programming B*, *45*(3), 503–528.

Lu, P. (1995). Optimal predictive control of continuous nonlinear systems. *International Journal of Control*, *62*(3), 633–649.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*(6), 789–814.

Meadows, E. S., & Rawlings, J. B. (1997). Model predictive control. In M. A. Henson, & D. E. Seborg (Eds.), *Nonlinear process control* (pp. 233–310). Englewood Cliffs, NJ: Prentice-Hall (Chapter 5).

Miele, A., Mangiavacchi, A., & Aggarwal, A. K. (1974). Modified quasi-linearization algorithm for optimal control problems with nondifferential constraints. *Journal of Optimization Theory and Applications*, *14*(5), 529–556.

Nash, S. G. (1984a). http://www.netlib.org/opt/tn.

Nash, S. G. (1984b). Newton-type minimization via the Lanczos algorithm. *SIAM Journal of Numerical Analysis*, *21*(4), 770–788.

Nocedal, J. (1990). http://www.netlib.org/opt/lbfgs_um.shar.

Ohtsuka, T. (1998). Time-variant receding-horizon control of nonlinear systems. *Journal of Guidance, Control, and Dynamics*, *21*(1), 174–176.

Ohtsuka, T. (2000). http://www-newton.mech.eng.osaka-u.ac.jp/∼ohtsuka/code/index.htm.

Ohtsuka, T., & Fujii, H. A. (1994). Stabilized continuation method for solving optimal control problems. *Journal of Guidance, Control, and Dynamics*, *17*(5), 950–957.

Ohtsuka, T., & Fujii, H. A. (1997). Real-time optimization algorithm for nonlinear receding-horizon control. *Automatica*, *33*(6), 1147–1154.

Qin, S. J., & Badgwell, T. A. (2000). An overview of nonlinear model predictive control applications. In F. Allgöwer, & A. Zheng (Eds.), *Nonlinear model predictive control* (pp. 369–392.) Basel: Birkhäuser.

Richter, S. L., & DeCarlo, R. A. (1983). Continuation methods: Theory and applications. *IEEE Transactions on Automatic Control*, *AC-28*(6), 660–665.

Tousain, R. L., & Bosgra, O. H. (2000). Efficient dynamic optimization for nonlinear model predictive control—application to a high-density poly-ethylene grade change problem. In *Proceedings of the 39th IEEE conference on decision and control*, December, 2000, Sydney, Australia (pp. 760–765).

Toshiyuki Ohtsuka was born in Tokyo, Japan, in 1967. He received his doctoral degree in aerospace engineering from the Tokyo Metropolitan Institute of Technology, Tokyo, Japan, in 1995. During 1995–1999, he was an Assistant Professor at the University of Tsukuba, Ibaraki, Japan. Since 1999, he has been with the Department of Computer-Controlled Mechanical Systems at the Graduate School of Engineering, Osaka University, Osaka, Japan, where he is currently an Associate Professor. His research interests include nonlinear control theory with applications to aerospace engineering and mechanical engineering.