Receding horizon control for spatiotemporal dynamic systems

Tomoaki HASHIMOTO*, Ryuta SATOH** and Toshiyuki OHTSUKA***

* Department of Mechanical Engineering, Osaka Institute of Technology
5-16-1 Omiya, Asahi-ku, Osaka, 535-8585, Japan
E-mail: info@thashi.net
** SoftBank Corporation
1-9-1 Higashi-shimbashi, Minato-ku, Tokyo 105-7303, Japan
*** Department of Systems Science, Kyoto University
Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan

Received 28 June 2015

Abstract
Receding horizon control is a type of optimal feedback control in which control performance over a finite future is optimized with a performance index that has a moving initial time and terminal time. Spatiotemporal dynamic systems characterized by both spatial and temporal variables often occur in many research fields. In this study, we develop a novel design method of receding horizon control for a generalized class of spatiotemporal dynamic systems. Using the variational principle, we first derive the exact stationary conditions that must be satisfied for a performance index to be optimized. Next, we provide a numerical algorithm to solve the stationary conditions via a finite-dimensional approximation. Finally, the effectiveness of the proposed method is verified by numerical simulations.

Key words: Control systems, Synthesis design, Optimal control, Spatiotemporal dynamics, Numerical solution

1. Introduction

Spatiotemporal dynamic systems described by partial differential equations (PDEs) often occur in many fields such as physics, biology, chemistry, and economics. The control of spatiotemporal dynamic systems is a challenging problem that arises in many fields of research.

Control of linear PDEs has been well established using the semigroup theory (Engel and Nagel, 2000). For a class of nonlinear PDEs, a control method, called the backstepping method, has been developed to achieve asymptotic stabilization in an infinite-dimensional setting (Krstic and Smyshlyaev, 2008). The key concept of the backstepping method is that variable transformation is used along with boundary feedback control to transform a nonlinear PDE into linear PDE. However, the backstepping method is inapplicable to a class of nonlinear PDEs whose coefficients have a nonlinear dependence on the state variable.

Receding horizon control (RHC), also known as model predictive control, is a well-established control method in which the current control input is obtained by solving a finite-horizon open-loop optimal control problem using the current state of the system as the initial state, and this procedure is repeated at each sampling instant. Thus, RHC is a type of optimal feedback control in which the control performance over a finite future is optimized with a performance index that has a moving initial time and terminal time. An important advantage of RHC is its ability to deal with constraints on control inputs and states, making it one of the most successful control methodologies because it enables control performance to be optimized while considering any constraints on state and control variables.

Recently, we proposed a methodology for designing receding horizon controllers for a particular class of one-dimensional nonlinear PDEs with state-dependent coefficients. It was shown in (Hashimoto, et al., 2013a) that the C/GMRES algorithm (Ohtsuka, 2004) can be applied to solve the RHC problem for a one-dimensional nonlinear thermal...
diffusion equation. Motivated by the fact that the optimality conditions for a class of RHC problem of nonlinear PDEs have a particular structure with respect to optimized parameters, we developed an efficient algorithm called the contraction mapping method (Hashimoto, et al., 2013b) as an alternative to the C/GMRES algorithm to numerically solve optimality conditions. It was shown in (Hashimoto, et al., 2013c) that the contraction mapping method can be applied to solve the RHC problem for high-dimensional Burgers’ equations. However, all the aforementioned methods cannot be applied to high-dimensional PDEs that contain high-order partial derivatives. Accordingly, the control of spatiotemporal dynamic systems is still an open problem as far as general classes of systems are concerned.

The objective of this study is to provide a generalized methodology for designing a receding horizon controller for high-dimensional spatiotemporal dynamic systems with high-order partial derivatives subject to constraints. The method proposed here is advantageous because of its applicability to a wide class of spatiotemporal dynamic systems. To verify the effectiveness of the proposed method, we address the application of the proposed method to the control problem of thermal fluid systems.

This paper is organized as follows. In Section 2, we introduce some notations and define the system model. In Section 3, we consider the RHC problem for a generalized class of spatiotemporal dynamic systems with boundary control inputs. Using the variational principle, we derive the stationary conditions that must be satisfied for a performance effectiveness of the proposed method. For a matrix $A \in \mathbb{R}^{n \times n}$, the transpose of $A$ is denoted by $A'$. Let $n_i, n_j, n_k, n_m, n_c \in \mathbb{N}_+$ and $h \in \mathbb{R} \setminus \{0\}$ be positive constants. Furthermore, let $m_x, m_y, m_z, m_t, m_e \in \mathbb{N}_+$ be positive constants. Let $x = [x_1, \ldots, x_n]^T \in \mathbb{R}^n$ and $t \in \mathbb{R}_+$ denote a spatial vector and temporal variable, respectively. Without loss of generality, we restrict our attention to the range $0 \leq x_i \leq h$ for all $i = 1, \ldots, n$. Let $\Omega$ and $\partial \Omega_i$ be the sets defined by $\Omega := \bigcap_{i=1}^{n} \{ x_i \mid 0 \leq x_i \leq h \}$ and $\partial \Omega_i := \{ x_i \mid x_i = 0, h \} \cap \Omega$, respectively. Let $z(x, t) = [z_1(x, t), \ldots, z_{n_i}(x, t)]' : \Omega \times \mathbb{R}_+ \rightarrow \mathbb{R}^{n_i}$ be a continuous vector-valued function with respect to $x$ and $t$. For $i = 1, \ldots, n_i$ and $j = 1, \ldots, n_d$, we introduce the following notations:

$$z_i(x, t) := \frac{\partial z(x, t)}{\partial t} := \left[ \frac{\partial z_1(x, t)}{\partial t}, \ldots, \frac{\partial z_{n_i}(x, t)}{\partial t} \right]' ,$$

$$z_{x_i}(x, t) := \frac{\partial z(x, t)}{\partial x_i} := \left[ \frac{\partial z_1(x, t)}{\partial x_i}, \ldots, \frac{\partial z_{n_i}(x, t)}{\partial x_i} \right]' ,$$

$$z_{x_i^j}(x, t) := \frac{\partial^2 z(x, t)}{\partial x_i \partial x_j^l} := \left[ \frac{\partial^2 z_1(x, t)}{\partial x_i \partial x_j^l}, \ldots, \frac{\partial^2 z_{n_i}(x, t)}{\partial x_i \partial x_j^l} \right]' ,$$

$$z_{x_i, x_j^l}(x, t) := \frac{\partial z(x, t)}{\partial x_i \partial x_j^l} := \left[ \frac{\partial z_i(x, t)}{\partial x_i \partial x_j^l} \right] ,$$

$$\int_{\Omega} \cdot \, \text{d}x := \int_0^h \cdots \int_0^h \cdot \, \text{d}x_1 \, \text{d}x_2 \cdots \text{d}x_{n_i} ,$$

$$\int_{\partial \Omega_i} \cdot \, \text{d}x := \int_0^h \cdots \int_0^h \cdot \, \text{d}x_{i-1} \, \text{d}x_{i+1} \cdots \text{d}x_{n_i} ,$$

$$s(x_i) := \begin{cases} 1 & \text{for } x_i = h, \\ -1 & \text{for } x_i = 0. \end{cases}$$

Let $\mathbb{D}$ be the set defined by $\mathbb{D} := \{ z(x, t) : \Omega \times \mathbb{R}_+ \rightarrow \mathbb{R}^{n_i} \mid z_i(x, t) \text{ and } z_{x_i}(x, t) \text{ exist for all } i = 1, \ldots, n_i, \, j = 1, \ldots, n_d, \, x \in \Omega, \, t \in \mathbb{R}_+ \}$. Let $z(x, t) \in \mathbb{D}$ and $u_i(x, t) : \partial \Omega_i \times \mathbb{R}_+ \rightarrow \mathbb{R}^{n_i} (i = 1, \ldots, n_i)$ be the state vector and the control input, respectively. Here, we consider the following spatiotemporal dynamic system written in a general setting:

$$\frac{\partial z(x, t)}{\partial t} = A \left( z, z_{x_1}, z_{x_1^2}, \ldots, z_{x_j}, \ldots \right) , \quad (1)$$

with boundary inputs

$$\frac{\partial z(x, t)}{\partial x_i} = u_i(x, t) \text{ for } x \in \partial \Omega_i , \quad (2)$$

[DOI: 10.1299/mej.15-00345] © 2016 The Japan Society of Mechanical Engineers
boundary conditions
\[ B \left( z, z_i, z_i^2, \cdots, z_i^\ell \right) = 0, \quad \text{for } x \in \partial \Omega, \]

and the initial condition \( z(0) = z_0(x) \). Here \( A \) and \( B \) are \( n_i \)- and \( n_r \)-dimensional vector-valued functions, respectively. Therein, \( z_0(x) : \Omega \to \mathbb{R}^n \) denotes the initial state satisfying Eqs. (1)–(4). Furthermore, the equality constraints are imposed as
\[ C \left( u, z, z_i, z_i^2, \cdots, z_i^\ell \right) = 0, \]

where \( C \) is an \( n_i \)-dimensional vector-valued function. It is well known that an inequality constraint can be converted into an equality constraint by introducing a slack variable (Bryson and Ho, 1975). Moreover, note that the boundary conditions specified in Eq. (3) can be included in Eq. (4).

The existence and regularity of a solution, the so-called Cauchy problem, are beyond the scope of this study. Thus, we assume that the solution of Eqs. (1)–(4) is unique and sufficiently smooth. Hereafter, we assume that \( z(x, t) \) is known at the present time \( t \) for all \( x \in \Omega \).

For example, mathematical models of the cooling process of a hot strip mill (Kumar, et al., 2001), magnetohydrodynamic turbulent flow (Muller and Buhler, 2001), long waves in shallow water (Balogh and Krstic, 2000), and quantum dynamics (Krstic, et al., 2011) belong to a class of system defined by Eq. (1). By transforming hyperbolic PDEs containing \( \frac{\partial z(x, t)}{\partial t} + \frac{\partial^2 z(x, t)}{\partial x^2} \) into a form of Eq. (1) by introducing an augmented state vector such as \( \tilde{z}(x, t) := \left[ z'(x, t), (\partial z(x, t)/\partial t)' \right]' \), we observe that system models of the Timoshenko beam, Euler–Bernoulli beam, and transmission wave equation can be reduced to a class of system defined by Eq. (1). Therefore, developing a control methodology for a generalized class of system, as defined by Eq. (1), is a challenging problem that arises in many fields of research.

3. Receding Horizon Control

In this section, we consider the receding horizon control problem for a class of system defined by Eq. (1). Using the variational principle, we analytically derive the stationary conditions that must be satisfied for a performance index to be optimized. For this purpose, we exploit integration by parts, a procedure that plays an important role in this study.

The control input at each time \( t \) is determined so as to minimize the performance index given by
\[ J = \int_\Omega \varphi(z(x, t + T))dx + \int_t^{t+T} \int_\Omega L(z(x, \tau), u(x, \tau))dx d\tau, \]

where \( T \in \mathbb{R} \) is the evaluation interval of the performance index, \( \varphi \in \mathbb{R}_+ \) is the terminal cost function, and \( L \in \mathbb{R}_+ \) is the cost function over the prediction horizon. The horizon \( T \) may vary with time, \( T = T(t) \), in general. The optimization problem of Eq. (5) subject to Eq. (1) can be reduced to the minimization of the following performance index introduced using the costate \( \lambda(x, t) \in \mathbb{R}^n \) and Lagrange multiplier \( \mu(x, t) \in \mathbb{R}^n \) associated with system equation in Eq. (1) and equality constraint in Eq. (4), respectively:
\[ J = \int_\Omega \varphi(z(x, t + T))dx + \int_t^{t+T} \int_\Omega \left( H - \lambda \frac{\partial z}{\partial \tau} \right) dx d\tau. \]

\( H \in \mathbb{R} \) denotes the Hamiltonian defined by
\[ H \left( z, z_i, u, \lambda, \mu \right) := L + \lambda' A + \mu' C. \]

Let \( \delta J, \delta z, \delta z_i, \delta \mu, \delta \lambda, \) and \( \delta \mu \) denote the variations (infinitesimal changes) in \( J, z, z_i, u, \lambda, \) and \( \mu, \) respectively. Note that we must perform the following integration by parts.
\[ \int_t^{t+T} \int_\Omega \left( A'(x, \tau) \frac{\partial z(x, \tau)}{\partial \tau} \right) dx d\tau = \left[ \int_\Omega A'(x, \tau) \delta z(x, \tau) dx \right]_t^{t+T} + \int_t^{t+T} \int_\Omega \frac{\partial \lambda(x, \tau)}{\partial \tau} \delta z(x, \tau) dx d\tau \]
\[ = \left[ \int_\Omega A'(x, t + T) \delta z(x, t + T) dx \right] + \int_t^{t+T} \int_\Omega \frac{\partial \lambda(x, \tau)}{\partial \tau} \delta z(x, \tau) dx d\tau. \]
In the above equation, we set $\delta z(x, t) = 0$ because $z(x, \tau)$ is fixed at $\tau = t$ as the present state. The above integration by parts can be used to convert $\delta z_i$ into $\delta z$. In addition, note that we can apply the following integration by parts procedure for the computation of $\delta f$.

$$\int_\Omega \frac{\partial H}{\partial z_i} \delta z_i \, dx = \int_\Omega \frac{\partial H}{\partial z_i} \delta z_i \, dx - \int_\partial \frac{\partial H}{\partial z_i} \delta z_i \, ds.$$  \hspace{1cm} (9)

$$\int_\Omega \frac{\partial^2 H}{\partial z_i \partial z_j} \delta z_i \, dx = \int_\Omega \frac{\partial^2 H}{\partial z_i \partial z_j} \delta z_i \, dx + \int_\partial \left( \frac{\partial H}{\partial z_i} \frac{\partial}{\partial z_j} \delta z_i \right) \delta z_j \, ds.$$  \hspace{1cm} (10)

$$\int_\Omega \frac{\partial H}{\partial z_i} \delta z_i \, dx = (-1)^j \int_\Omega \left( \frac{\partial^{j+1}}{\partial z_j^{j+1}} \frac{\partial H}{\partial z_i} \right) \delta z_i \, dx + \int_\partial \sum_{k=1}^j \left( (-1)^{j-k} \frac{\partial^{j-k}}{\partial z_j^{j-k}} \frac{\partial H}{\partial z_i} \right) \delta z_i \, ds.$$  \hspace{1cm} (11)

From boundary condition Eq. (2), we have

$$\delta z_i(x, \tau) = \delta u_i(x, \tau) \quad \text{for} \quad x \in \partial \Omega.$$  \hspace{1cm} (12)

Substituting Eq. (12) into the term $\delta z_i$ in Eq. (10) for $x \in \partial \Omega$, we obtain the following:

$$\int_\Omega \frac{\partial H}{\partial z_i} \delta z_i \, dx = \int_\Omega \frac{\partial^2 H}{\partial z_i \partial z_j} \delta z_i \, dx + \int_\partial \left( \frac{\partial H}{\partial z_i} \frac{\partial}{\partial z_j} \delta z_i \right) \delta z_j \, ds.$$  \hspace{1cm} (13)

For $j \geq 3$, substituting Eq. (12) into the term $\delta z_i$ in Eq. (11) for $x \in \partial \Omega$ yields

$$\int_\Omega \frac{\partial H}{\partial z_i} \delta z_i \, dx = (-1)^j \int_\Omega \left( \frac{\partial^{j+1}}{\partial z_j^{j+1}} \frac{\partial H}{\partial z_i} \right) \delta z_i \, dx$$
$$+ \int_\partial \sum_{k=1}^j \left( (-1)^{j-k} \frac{\partial^{j-k}}{\partial z_j^{j-k}} \frac{\partial H}{\partial z_i} \right) \delta z_i \, ds.$$  \hspace{1cm} (14)

Note that the variations $\delta z_i$ for $x \in \Omega$ and $j \in \{1, \ldots, n_d\}$ can be resolved into $\delta z_i$ for $x \in \Omega$, into $\delta u_i$ for $x \in \partial \Omega$, and into $\delta z_j$ for $x \in \partial \Omega$ and $j \in \{2, \ldots, n_d\}$. First, using the integration by parts in Eq. (8), we obtain the variation in $\delta J$ as follows:

$$\delta J = \int_\Omega \left( \frac{\partial \delta z}{\partial \mathbf{x}} \mathbf{z}(x, t + \tau) - \mathbf{X}(x, t + \tau) \right) \delta z \, dx + \int_\partial \left( \frac{\partial \delta z}{\partial \mathbf{x}} \mathbf{z}(x, t + \tau) - \mathbf{X}(x, t + \tau) \right) \delta z \, ds$$
$$+ \int_\Omega \left( \frac{\partial \delta \mathbf{u}}{\partial \mathbf{x}} \mathbf{u} \right) \delta \mathbf{u} \, dx + \int_\partial \left( \frac{\partial \delta \mathbf{u}}{\partial \mathbf{x}} \mathbf{u} \right) \delta \mathbf{u} \, ds,$$  \hspace{1cm} (15)

where $\delta H$ is given by

$$\delta H(z, z_i, u, \lambda, \mu) = \left( \frac{\partial H}{\partial z} \right) \delta z + \sum_{i=1}^{n_d} \left( \frac{\partial H}{\partial z_i} \right) \delta z_i + \sum_{i=1}^{n_d} \left( \frac{\partial H}{\partial u_i} \right) \delta u_i + \left( \frac{\partial H}{\partial \lambda} \right) \delta \lambda + \left( \frac{\partial H}{\partial \mu} \right) \delta \mu.$$  \hspace{1cm} (16)

Next, we apply integration by parts in Eqs. (9), (13), and (14) to the computation of $\delta H$ in Eq. (16).

$$\int_\Omega \left( \frac{\partial H}{\partial z_j} \right) \delta z_j \, dx = \int_\Omega \left( \delta z_j \mathbf{A} + \delta u_j \mathbf{C} \right) \delta z_j \, dx$$
$$+ \sum_{i=1}^{n_d} \int_\partial \left( \delta z_i \frac{\partial \mathbf{A}}{\partial \mathbf{x}} + \delta u_i \frac{\partial \mathbf{C}}{\partial \mathbf{x}} \right) \delta z_j \, ds.$$  \hspace{1cm} (17)

For the last term in Eq. (17), we have

$$\int_\Omega \left( \frac{\partial H}{\partial u_i} \right) \delta u_i \, dx = \int_\partial \left( \sum_{k=1}^{n_d} \left( \frac{\partial H}{\partial u_i} \right) \delta z_i \right) \delta u_i \, ds.$$  \hspace{1cm} (18)

[DOI: 10.1299/mej.15-00345] © 2016 The Japan Society of Mechanical Engineers
Considering Eqs. (15)–(18), we obtain $\delta J$ as follows:

$$
\delta J = \int_{\Omega} \left\{ \frac{\partial \{ \mathcal{L}[z] \}}{\partial z} - \mathcal{L}(x, t + T) \right\} \delta z(x, t + T) \, dx
+ \int_{t}^{T} \left[ \int_{\Omega} \delta \lambda^{i} \left( \frac{\partial}{\partial \tau} - \frac{\partial}{\partial z} \right) \mathcal{L}(x, t + T) \, dx + \lambda^{i} \mathcal{L}(x, t + T) \right] \, dt
+ \sum_{i=1}^{n} \int_{\Omega_{i}} \sum_{x \in \partial \Omega_{i}} \left\{ \left( \frac{\partial H}{\partial u} + \sum_{j=2}^{n_{i}} s(x) (-1)^{j-2} \frac{\partial \iota^{j-2}}{\partial x^{j-2}_{i}} \frac{\partial H}{\partial z_{j-1}^{i}} \right) \delta u_{i} + s(x) \sum_{j=3}^{n_{i}} \sum_{k=0}^{n_{i} - j} (-1)^{k} \frac{\partial \iota^{k}}{\partial x_{j-1}^{i}} \frac{\partial H}{\partial z_{j-1}^{i}} \delta z_{j-1}^{i} \right\} \, dx \, dt.
$$

(19)

On the basis of the variational principle, we obtain the necessary conditions for a stationary value of $J$ over the horizon $(t \leq \tau \leq t + T)$ as follows. For $x \in \Omega$, we obtain

$$
\frac{\partial \mathcal{L}(x, \tau)}{\partial \tau} = \mathcal{A} \left( z(x, \tau), z_{x}, z_{x^{2}}, \ldots, z_{x^{l}}, \ldots \right),
$$

(20)

$$
\lambda(x, t + T) = \left\{ \frac{\partial \mathcal{L}[z(\mathcal{L}(x, t + T))]}{\partial z} \right\},
$$

(21)

$$
\left( \frac{\partial \lambda}{\partial \tau} \right) = \left\{ \frac{\partial H}{\partial u} - \sum_{i=1}^{n} \sum_{j=1}^{n_{i}} (-1)^{j} \frac{\partial \iota^{j-1}}{\partial x_{j-1}^{i}} \frac{\partial H}{\partial z_{j-1}^{i}} \right\},
$$

(22)

$$
\mathcal{C} \left( u, z, z_{x}, z_{x^{2}}, \ldots, z_{x^{l}}, \ldots \right) = 0.
$$

(23)

For $i = 1, \ldots, n_{i}$ and $x_{i} \in \partial \Omega_{i}$, we obtain

$$
\left( \frac{\partial H}{\partial u} + \sum_{j=2}^{n_{i}} s(x) (-1)^{j-2} \frac{\partial \iota^{j-2}}{\partial x^{j-2}_{i}} \frac{\partial H}{\partial z_{j-1}^{i}} \right) = 0,
$$

(24)

$$
\sum_{j=1}^{n_{i}} (-1)^{j-2} \frac{\partial \iota^{j-1}}{\partial x_{j-1}^{i}} \frac{\partial H}{\partial z_{j-1}^{i}} = 0.
$$

(25)

For $i = 1, \ldots, n_{i}$, $x_{i} \in \partial \Omega_{i}$, and $j = 3, \ldots, n_{i}$, we obtain

$$
\sum_{\ell=0}^{n_{i} - j} (-1)^{\ell} \frac{\partial \iota^{\ell}}{\partial x_{j-1}^{i}} \frac{\partial H}{\partial z_{j-1}^{i}} = 0.
$$

(26)

Conditions Eqs. (20)–(26) are called the stationary conditions and must be satisfied so that the performance index defined in Eq. (6) is minimized. A well-known difficulty in solving nonlinear optimal control problems is that the obtained stationary conditions cannot be solved analytically in general.

4. Numerical Solution

Although we have analytically derived the exact stationary conditions in Section 3, we need a numerical algorithm to solve the stationary conditions. In this section, we provide a framework so that the fast on-line algorithms called the C/GMRES method (Ohtsuka, 2004) or contraction mapping method (Hashimoto, et al., 2013b) are applicable for solving the receding horizon control problem in nonlinear PDEs.

4.1. Procedure for Solving Stationary Conditions

Here note that the stationary conditions contain two time-evolutionary equations in Eqs. (20) and (22) with respect to $z$ and $\lambda$, respectively. The remaining equations are algebraic equations, and Eqs. (25)–(26) are considered as the boundary conditions for the time-evolutionary equation of $\lambda$. Let $\mathcal{U}$ be defined by $\mathcal{U} = [u_{i}, \ldots, u_{n_{i}}, \mu]$. For a given initial solution $\mathcal{U}(x, \tau), \tau \in [t, t + T]$ and the present state $z(x, t)$, we first determine $z(x, \tau)$ for $\tau \in [t, t + T]$ by numerically solving Eq. (20) with boundary conditions in Eqs. (2)–(3) from $\tau = t$ to $\tau = t + T$. Then, the terminal costate $\lambda(x, t + T)$ is determined from the obtained terminal state $z(x, t + T)$ by Eq. (21). Consequently, $\lambda(x, \tau)$ for $\tau \in [t, t + T]$ is also determined by numerically solving Eq. (22) with boundary conditions in Eq. (25)–(26) from $\tau = t + T$ to $\tau = t$. Figure 1 shows that the procedure for solving the time-evolutionary equation of $z$ is forward, whereas the one for solving the time-evolutionary equation of $\lambda$ is backward.
To solve stationary conditions in Eqs. (20)–(26) using a numerical algorithm, we must discretize equations in Eqs. (20)–(26) into finite difference equations. Let $x \in \Omega$ be divided into $m_s$ grid points and $\tilde{x} := [\tilde{x}_1, \ldots, \tilde{x}_{m_s}]^\top \in \mathbb{R}^{m_s}$ denote the discretized spatial vector. Likewise, let the time $\tau \in [t, t + T]$ over the prediction horizon be divided into $m_t$ steps and $\tilde{\tau} := [\tilde{\tau}_1, \ldots, \tilde{\tau}_{m_t}]^\top \in \mathbb{R}^{m_t}$ denote the discretized temporal vector. Note that $\tilde{\tau}_1$ is identical to the present time $t$. Let the set $\{\tilde{\delta}_1, \ldots, \tilde{\delta}_{m_t}\}$ be given by $[\tilde{x}_1, \ldots, \tilde{x}_{m_t}] \cap \partial \Omega_t$. Let $\tilde{\delta}\tilde{x}_k \in \mathbb{R}^{m_s}$ be defined by $\tilde{\delta}\tilde{x}_k := [\tilde{\delta}_1, \ldots, \tilde{\delta}_{m_t}]^\top$. Let $\tilde{u}(\tilde{\delta}\tilde{x}, \tilde{\tau}) := [\tilde{u}_1(\tilde{\delta}\tilde{x}_1, \tilde{\tau}_1), \ldots, \tilde{u}_{m_t}(\tilde{\delta}\tilde{x}_{m_t}, \tilde{\tau}_{m_t})]^\top$ denote the discretized control input. Let $\tilde{z}(\tilde{x}, \tilde{\tau}), \tilde{\lambda}(\tilde{x}, \tilde{\tau})$, and $\tilde{\mu}(\tilde{x}, \tilde{\tau})$ denote the discretized state, costate, and Lagrange multiplier, respectively. For notational simplicity, let $\hat{\lambda}_{k_1}$ be denoted by $\tilde{\lambda}_k$ for $k = 1, \ldots, m_t$. Note that $\hat{\lambda}_{k_1}$ is identical to the present known state $\hat{\lambda}(k_1)$. For other variables, we adopt similar notations without explanation. As a result of the finite difference approximation, we obtain the discretized stationary conditions over the horizon ($k = 1, \ldots, m_t$) as follows:

$$
\begin{align*}
\tilde{z}_{k+1} &= \tilde{\Lambda}(\tilde{z}_k, \tilde{u}_k), & (27) \\
\tilde{\lambda}_m &= \tilde{\Phi}(\tilde{\lambda}_m), & (28) \\
\tilde{\lambda}_k &= \tilde{\Phi}(\tilde{z}_{k+1}, \tilde{u}_{k+1}, \tilde{\mu}_{k+1}), & (29) \\
\tilde{C}_k(\tilde{u}_k, \tilde{z}_k) &= 0, & (30) \\
\tilde{E}_k(\tilde{u}_k, \tilde{z}_k, \tilde{\lambda}_k, \tilde{\mu}_k) &= 0, & (31)
\end{align*}
$$

Here $\tilde{\Lambda}$, $\tilde{\Phi}$, and $\tilde{D}$ are $m_s$-dimensional vector-valued functions, $\tilde{C}$ is an $m_s$-dimensional vector-valued function, and $\tilde{E}$ is an $m_s$-dimensional vector-valued function. The specific forms of these functions depend on the manner of discretization. We consider the discretized conditions in a general setting as shown above without referring to any specific discretization method. The time-evolutionary equations of $z$ and $\lambda$ are discretized in forward difference equation in Eq. (27) and backward difference equation in Eq. (29), respectively. Note that boundary conditions in Eqs. (2)–(3) are discretized and employed in Eq. (27). Moreover, the equations obtained by discretizing Eqs. (22), (25), and (26) are unified into Eq. (29). In addition, the remaining stationary conditions in Eqs. (21), (23), and (24) are discretized and described in general forms as Eqs. (28), (30), and (31), respectively.

For each time $t$, optimization parameters $\tilde{u}_k$ and $\tilde{\mu}_k$ are determined to satisfy the stationary conditions. Thereafter, only the first input $\tilde{u}_0$ is employed in the controlled object at real time $t$. The predictive horizon recedes as real time $t$ is increased by the sampling period $\Delta t$. To achieve real-time optimization, we must repeatedly solve stationary conditions in Eqs. (27)-(31) within the sampling period $\Delta t$.

For the present time $t$, let optimization parameters $\tilde{u}_k$ and $\tilde{\mu}_k$ for $k = 1, \ldots, m_t$ be combined into the vector defined by

$$
\tilde{U}(t) := [\tilde{u}_1', \ldots, \tilde{u}_m', \tilde{\mu}_1', \ldots, \tilde{\mu}_m']'.
$$

For the present state $\tilde{z}_1(t)$ and a given initial solution $\tilde{U}(t)$, $\tilde{z}_k(t)$ for $k = 1, \ldots, m_t$ is calculated recursively from $k = 1$ to $k = m_t$, by Eq. (27). Next, the terminal costate $\tilde{\lambda}_m(t)$ is determined from the terminal state $\tilde{z}_m(t)$ by Eq. (28). Consequently, $\tilde{\lambda}_k(t)$ for $k = 1, \ldots, m_t$ is calculated recursively from $k = m_t$ to $k = 1$ by Eq. (29). Because $\tilde{z}_k(t)$ and $\tilde{\lambda}_k(t)$ are determined by $\tilde{z}_1(t)$ and $\tilde{U}(t)$ through Eqs. (27)–(29), Eqs. (30)–(31) can be regarded as a single equation,

$$
\text{F} \left( \tilde{U}(t), \tilde{z}_1(t), t \right) := \left[ \tilde{C}_1', \ldots, \tilde{C}_m', \tilde{E}_1', \ldots, \tilde{E}_m' \right]' \in \mathbb{R}^{m_r},
$$

where $m_r := m, m_t + m, m_t$. Because $\tilde{z}_k(t)$ and $\tilde{\lambda}_k(t)$ are uniquely determined through Eqs. (27)–(29) for the given $\tilde{z}_1(t)$ and $\tilde{U}(t)$, $\tilde{z}_k(t)$ and $\tilde{\lambda}_k(t)$ depend on $\tilde{z}_1(t)$ and $\tilde{U}(t)$. Hence, it is reasonable to consider the arguments of $\text{F}$ as $\tilde{U}(t), \tilde{z}_1(t), t$.

For a given $\tilde{z}_1(t)$ and $\tilde{U}(t)$, $\text{F}$ is not necessarily equal to zero, so $\|\text{F}\|$ is used to evaluate the optimality performance. If $\|\text{F}\| = 0$ is satisfied for the given $\tilde{z}_1(t)$ and $\tilde{U}(t)$, then the stationary conditions are satisfied. Several algorithms have been developed such that $\|\text{F}\|$ can be decreased by suitably updating $\tilde{U}(t)$, as discussed below.
4.2. C/GMRES Method

A conventional way to update \( \hat{U}(t) \) is to replace \( \hat{U}(t) \) with \( \hat{U}(t) + \alpha s \), known as the steepest descent method, where \( s \) is the steepest descent direction and \( \alpha \) is the step length satisfying the Armijo condition (Nocedal and Wright, 2006). For Newton’s method, \( s \) is given by the Hessian instead of the gradient. However, these methods are computationally expensive, and it was shown that the C/GMRES algorithm (Ohtsuka, 2004) is not only faster but also more numerically robust than these conventional algorithms. Next, a brief description of the C/GMRES method applied to this problem is provided. Instead of solving \( F(\hat{U}(t), \hat{z}(t), t) = 0 \) itself at each time by an iterative method such as the steepest descent method or Newton’s method, we find the derivative of \( \hat{U}(t) \) with respect to time so that \( F(\hat{U}(t), \hat{z}(t), t) = 0 \) is satisfied identically. Namely we determine \( \hat{U}(t) \) such that

\[
F(\hat{U}(t), \hat{z}(t), t) = -\varepsilon F(\hat{U}(t), \hat{z}(t), t)
\]

is satisfied, where \( \varepsilon \) is a positive constant introduced to stabilize \( F = 0 \). If we choose \( \varepsilon = 1/\Delta t \), then the stability of Eq. (33) with forward difference approximation is guaranteed (Ohtsuka, 2004), where \( \Delta t \) denotes the sampling period. By total differentiation, we obtain

\[
\frac{\partial F}{\partial \hat{U}(t)} \hat{U}(t) = -\varepsilon F - \frac{\partial F}{\partial \hat{z}_1} \hat{z}_1 - \frac{\partial F}{\partial t}.
\]

This equation can be regarded as a linear algebraic equation with the coefficient matrix \( (\partial F/\partial \hat{U}(t)) \), which can be used to determine \( \hat{U} \) for the given \( \hat{U}, \hat{z}_1, \hat{z}_2, \) and \( t \). Then, if the Jacobian \( (\partial F/\partial \hat{U}) \) is nonsingular, we obtain the following differential equation for \( \hat{U}(t) \):

\[
\hat{U}(t) = \left( \frac{\partial F}{\partial \hat{U}(t)} \right)^{-1} \left( -\varepsilon F - \frac{\partial F}{\partial \hat{z}_1} \hat{z}_1 - \frac{\partial F}{\partial t} \right)
\]

(35)

We can update the solution \( \hat{U}(t) \) of \( F(\hat{U}(t), \hat{z}(t), t) = 0 \) without using an iterative optimization method by integrating Eq. (35) in real time as, for example, \( \hat{U}(t + \Delta t) = \hat{U}(t) + \hat{U}(t)\Delta t \). This approach is a type of continuation method (Richter and DeCarlo, 1983) in the sense that the solution curve \( \hat{U}(t) \) is traced by integrating a differential equation. From the computational viewpoint, the differential equation in Eq. (35) still involves expensive operations, i.e., solving the Jacobians \( (\partial F/\partial \hat{U}(t)), (\partial F/\partial \hat{z}_1), \) and \( (\partial F/\partial t) \) and linear algebraic equation associated with \( (\partial F/\partial \hat{U})^{-1} \). To reduce the computational cost of the Jacobians and linear equation, we employ two techniques: the forward difference approximation for the products of the Jacobians and vectors and the GMRES method (Kelley, 1995) for the linear algebraic equation. Using the forward difference approximation, we can obtain a linear equation with respect to \( \hat{U} \). Thereafter, we can apply the GMRES algorithm to find the solution \( \hat{U}(t) \) of the linear equation. Consequently, \( \hat{U} \) can be updated so that \( F = 0 \) is stabilized. More detailed information about the implementation of C/GMRES is provided in (Ohtsuka, 2004).

4.3. Contraction Mapping Method

Recently, we have developed a more efficient algorithm than C/GMRES, called the contraction mapping method (Hashimoto et al., 2013b). In particular, we have proposed an algorithm for solving \( F(U, \hat{z}_1, t) = 0 \) under the assumption that \( F(U, \hat{z}_1, t) \) satisfies a particular structural condition with respect to \( U \). Compared with the C/GMRES method, the contraction mapping method has the disadvantage of limited applicability but the advantage of a smaller computational burden. Next, a brief description of the contraction mapping method is provided. Hereafter, we assume that \( F \) is given by

\[
F(U(t), \hat{z}_1(t), t) = QU(t) - R(U(t), \hat{z}_1(t), t),
\]

(36)

where \( Q \in \mathbb{R}^{m \times m'} \) is a nonsingular constant matrix and \( R \in \mathbb{R}^{m'} \) is a vector-valued function satisfying the following Lipschitz continuous conditions: for \( v_1, v_2, v_3, v_4 \in \mathbb{R}^m \),

\[
\|R \left( \hat{U}_{(t+\Delta t)}, \hat{z}_{(t+\Delta t)}, t + \Delta t \right) - R \left( \hat{U}_{(t)}, \hat{z}_{(t)}, t \right) \| \leq v_1 \| \hat{U}_{(t+\Delta t)} - \hat{U}_{(t)} \| + v_2 \| \hat{z}_{(t+\Delta t)} - \hat{z}_{(t)} \| + v_3 \Delta t,
\]

\[
\| \hat{z}_{(t+\Delta t)} - \hat{z}_{(t)} \| \leq v_4 \Delta t.
\]

Let \( P \in \mathbb{R}^{m \times m} \) be defined by

\[
P(U, \hat{z}_1, t) = Q^{-1}R(U, \hat{z}_1, t).
\]

(37)
5. Illustrative example

In this section, we provide an illustrative example to verify the effectiveness of the proposed method. For the two-dimensional square domain \( \Omega := [0, 1] \times [0, 1] \), we consider an incompressible flow of thermal fluid dynamics that can be described by momentum, continuity, and energy equations. Here the momentum equation is given by the Navier–Stokes equation with the Boussinesq approximation.

\[
\begin{align*}
\nabla \cdot \mathbf{v} &= 0, \\
\frac{\partial \mathbf{v}}{\partial t} &= -\frac{1}{\rho} \nabla P - (\mathbf{v} \cdot \nabla) \mathbf{v} + \nu \nabla^2 \mathbf{v} + \mathbf{g} \beta (\theta - \theta_0),
\end{align*}
\]

(45)

where \( \mathbf{v} \) is the velocity \([\text{m/s}]\), \( \theta(x, t) \in \mathbb{R} \) is the temperature \([\text{K}]\), and \( P(x, t) \in \mathbb{R} \) is the pressure \([\text{N/m}^2]\), and they are considered as the state vector \( \mathbf{z} := [\mathbf{v}', \theta']', \nabla, \nabla^2 \), and \( \mathbf{v} \cdot \nabla \) are given by \( \mathbf{v} := \left( \frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2} \right)' \), \( \nabla^2 := \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} \), and \( \mathbf{v} \cdot \nabla := v_1 \frac{\partial}{\partial x_1} + v_2 \frac{\partial}{\partial x_2} \), respectively. The other notations in Eqs. (45)–(47) are all constant parameters as follows: \( \rho \in \mathbb{R} \) is the density \([\text{kg/m}^3]\), \( \nu \in \mathbb{R} \) is the kinematic viscosity \([\text{m}^2/\text{s}]\), \( \mathbf{g} \in \mathbb{R}^2 \) is the gravity acceleration \([\text{m/s}^2]\), \( \theta_0 \) is the reference temperature \([\text{K}]\), \( \alpha \) is the thermal diffusivity coefficient \([\text{m}^2/\text{s}]\), and \( \beta \) is the thermal expansion coefficient \([1/\text{K}]\). Here we consider the following boundary conditions:

\[
\begin{align*}
\mathbf{v} &= 0, \quad \theta_{\Omega_1} = u_1, \quad P_{\Omega_1} = 0, \quad \text{for } x \in \partial \Omega_1, \\
\mathbf{v} &= 0, \quad \theta_{\Omega_2} = u_2, \quad P_{\Omega_2} = 0, \quad \text{for } x \in \partial \Omega_2,
\end{align*}
\]

(48)

and the initial conditions: \( \mathbf{v}(x, 0) = 0, \quad \theta(x, 0) = \theta_0(x) \).

Consider a temperature control problem of air flow that is governed by Eqs. (45)–(47). The gradient of the temperature in the boundary region is regarded as the control input. Considering the fact that the controlled object is the air flow, we set the system parameters as follows: \( \rho = 1.25, \quad v = 1.38 \times 10^{-5}, \quad \mathbf{g} = [0, 9.8]' \), \( \theta_0 = 300, \quad \alpha = 1.91 \times 10^{-5} \), and \( \beta = 38 \).
\( \beta = 3.33 \times 10^{-3} \). The desired temperature \( \theta_f(x) \) is set as \( \theta_f(x) = 310 \) for all \( x \in \Omega \). Here, we introduce the following performance index defined by

\[
\varphi = \frac{1}{2} w_1 \left( \theta(x, t + T) - \theta_f \right)^2,
\]

\[
L = \frac{1}{2} w_2 \left( \theta(x, \tau) - \theta_f \right)^2 + w_3 \sum_{i=1}^{2} u_i^2(x, \tau).
\]

Considering that \( A \in \mathbb{R}^3 \) and \( C \in \mathbb{R} \) are vector-valued and scalar functions, respectively, given by

\[
A := \left( \begin{array}{c} -\frac{1}{2} \nabla z_4 - G \left( \begin{array}{c} z_1 \\ z_2 \end{array} \right) + g \beta (z_3 - \theta_1) \\ \nabla \\ \nabla \end{array} \right) z_3 + \alpha \nabla^2 z_3,
\]

\[
C := \nabla \cdot \left( \begin{array}{c} z_1 \\ z_2 \end{array} \right),
\]

we obtain the Hamiltonian as shown in Eq. (7) by introducing the costate \( \lambda \in \mathbb{R}^3 \) and Lagrange multiplier \( \mu \in \mathbb{R} \). Applying the formulation in Eqs. (20)-(26), we obtain the following stationary conditions. For \( x \in \Omega \), we have

\[
\frac{\partial}{\partial t} \left[ z_1, z_2, z_3 \right] \mu = A(z),
\]

\[
\nabla \cdot \left[ \begin{array}{c} z_1 \\ z_2 \end{array} \right] = 0,
\]

\[
\left[ \begin{array}{c} A(x, t + T) \\ \mu(x, t + T) \end{array} \right] = \left[ \begin{array}{c} 0, 0, w_1 (z_3(x, t + T) - \theta_f) \end{array} \right],
\]

\[
\frac{\partial}{\partial t} \left[ \begin{array}{c} \lambda_1, \lambda_2, \lambda_3 \end{array} \right] = D(z, \lambda, \mu),
\]

\[
\nabla \cdot \left[ \begin{array}{c} \lambda_1 \\ \lambda_2 \end{array} \right] = 0.
\]

For \( i = 1, 2 \) and \( x_i \in \partial \Omega \), we have

\[
w_3 \mu_i (x, \tau) + s(x_i) \alpha \lambda_3 (x, \tau) = 0,
\]

\[
\left[ \begin{array}{c} \lambda_1 \\ \lambda_2 \end{array} \right] = 0, \quad \frac{\partial \lambda_3}{\partial x_i} = 0.
\]

\( D \in \mathbb{R}^3 \) is given by

\[
D := \left( \nabla \mu + \left( \nabla \cdot \left( \begin{array}{c} z_1 \\ z_2 \end{array} \right) \right) \left( \begin{array}{c} \lambda_1 \\ \lambda_2 \end{array} \right) - H + \lambda_3 \nabla z_3 \right),
\]

where \( H \in \mathbb{R}^2 \) and \( I \in \mathbb{R} \) are given by

\[
H := \left( \begin{array}{c} z_1 \\ z_2 \end{array} \right) \cdot \lambda_1 + \lambda_2 \left[ \begin{array}{c} z_1 \\ z_2 \end{array} \right] + \lambda_3 \nabla z_3,
\]

\[
I := \beta \left( \begin{array}{c} \lambda_1 \\ \lambda_2 \end{array} \right) + w_2 \left( z_3 - \theta_f \right).
\]

In the following, we provide a brief description of the numerical method for solving stationary conditions (55)-(61). Several numerical algorithms are applicable to solving Boussinesq equations (55)-(56). In general, the solution obtained by simply integrating time-evolutionary equation (55) does not satisfy continuity equation (56). Note that \( A(z) \) in (52)
contains unknown variable $z_4$. Hence, we can consider $z_4$ as the flexible parameter to be adjusted for satisfying continuity equation (56). We adopt here the simplified marker and cell (SMAC) method (Amsden and Harlow, 1970), in which $z_4$ is updated through the integration of (55) so as to satisfy (56). Hence, using the SMAC method, we can determine $z(x, \tau)$ over the prediction horizon ($t \leq \tau \leq t + T$) from $\tau = t$ to $\tau = t + T$.

Furthermore, the terminal costate $\lambda(t + T)$ is determined by (57). Note that there is a duality between equations (55)-(56) and (58)-(59). Hence, the SMAC method also can be applied to solve (58)-(59) with boundary conditions (61). Consequently, for the present state $z(x, \tau)$ and a given $U = [u_1, u_2]'$, $z(x, \tau)$ and $\lambda(x, \tau)$ are determined over the prediction horizon ($t \leq \tau \leq t + T$).

Thus, the optimization problem can be reduced to solving single condition (60). In fact, we can apply not only the C/GMRES method in Section 4.2 but also the contraction mapping method in Section 4.3 for solving optimality condition (60), because condition (60) can be reduced into the form in (36).

First, we solve the optimization problem by applying the contraction mapping method, in which we adopt a staggered grid (Chung, 2010) and the SMAC method for numerical computation of the time-evolutionary equations for $z$ and $\lambda$. Next, we solve the same optimization problem by applying the C/GMRES method for comparing each performance.

Simulation is performed on a laptop computer (CPU: Intel(R) Core(TM) i5-4200U 1.6 [GHz], Memory: 8.0 [GB], OS: Windows 8, Software: Matlab). The average computational times per update (one control cycle) are 1.31 [s] and 6.32 [s] by the contraction mapping method and the C/GMRES method, respectively.

In the following, we provide the simulation results to verify the effectiveness of the proposed method. The parameters employed in the numerical simulations are as follows: $\Delta t = 0.02$, $m_x = 30 \times 30$, $m_t = 10$, $[w_1, w_2, w_3] = [10^5, 10^6, 1]$, and $k = 1$. Owing to the initialization of the optimal solution $U(0)$, the length of the horizon is selected so that $T(0) = 0$ and $T(t) \rightarrow 0.1$ as $t \rightarrow \infty$, i.e., $T = 0.1(1 - e^{-0.5t})$.

In Figs. 2–15, even and odd numbers show the time response of temperature $\theta$ and flow velocity $v$, respectively, controlled by receding horizon control via the contraction mapping method. These figures reveal the effectiveness of the proposed method. Since the time response of the state controlled by receding horizon control via the C/GMRES method is almost same, it is omitted. Figures 17–19 show the time response of the control inputs. Figures 20 and 21 show the time response of the state and optimality errors, respectively. It is apparent that the state and optimality errors converge to zero. Moreover, it can be seen that the optimality error of the contraction mapping method is larger than that of the C/GMRES method, but the computational time of the contraction mapping method is smaller than that of the C/GMRES method. We can see that there is a tradeoff between computational burden and error performance.

Fig. 2 $\theta$ at $t = 0$

Fig. 3 $v$ at $t = 0$

Fig. 4 $\theta$ at $t = 2$

Fig. 5 $v$ at $t = 2$
Fig. 6  $\theta$ at $t = 4$

Fig. 7  $v$ at $t = 4$

Fig. 8  $\theta$ at $t = 6$

Fig. 9  $v$ at $t = 6$

Fig. 10  $\theta$ at $t = 8$

Fig. 11  $v$ at $t = 8$

Fig. 12  $\theta$ at $t = 10$

Fig. 13  $v$ at $t = 10$

Fig. 14  $\theta$ at $t = 40$

Fig. 15  $v$ at $t = 40$
6. Conclusion

In this study, we provided a methodology to design a receding horizon controller for a generalized class of spatiotemporal dynamic systems subject to constraints. This method is advantageous for its applicability to a wide class of spatiotemporal dynamic systems. The temperature control of thermal fluid dynamics was examined to verify the effectiveness of the proposed method. However, the stability of the closed-loop system controlled by the proposed method is not theoretically guaranteed. Robust stability of the closed-loop system against modeling errors and disturbances should be guaranteed, which is a problem to be considered in future studies.

Acknowledgement

This work was supported in part by Japan Society for the Promotion of Science, Grant-in-Aid for Young Scientists (24760334) and Grant-in-Aid for Scientific Research (24360165 and 15H02257). This work was also partially supported by the Kurata Grant awarded by the Kurata Memorial Hitachi Science and Technology Foundation.

References