Receding Horizon Control with Numerical Solution for Spatiotemporal Dynamic Systems

Tomoaki Hashimoto, Yusuke Yoshioka and Toshiyuki Ohtsuka

***Abstract*— Receding horizon control is a kind of optimal feedback control, and its performance index has a moving initial time and a moving terminal time. Spatiotemporal dynamic systems are often described by partial differential equations, and their behavior is characterized by both spatial and temporal variables. In this study, we develop a 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 for solving the stationary conditions via finite-dimensional approximation. Finally, the effectiveness of the proposed method is verified by numerical simulations.**

I. INTRODUCTION

Spatiotemporal dynamic systems described by partial differential equations (PDEs) occur in many fields such as physics, biology, chemistry, and economics. Control of linear PDEs has been well established using semigroup theory [1]-[2]. For a class of nonlinear PDEs, a control method called the backstepping method has been well developed to achieve the asymptotic stabilization in an infinite-dimensional setting [3]. The key concept of the backstepping method is that variable transformation is used along with boundary feedback control to transform a nonlinear PDE into a linear PDE. However, the backstepping method is inapplicable to a class of nonlinear PDEs whose coefficients nonlinearly depend on the state variable. The control of spatiotemporal dynamic systems is still an open problem as far as general classes of systems are concerned.

Receding horizon control, also known as model predictive control, is a kind of optimal feedback control in which the control performance over a finite future is optimized and its performance index has a moving initial time and a moving terminal time. The research on receding horizon control can be classified into off-line optimization [4] and on-line optimization [5]. This study deals with the on-line optimization problem.

Recently, we have proposed a design method of receding horizon control for one-dimensional nonlinear parabolic PDEs with state-dependent coefficients [6]. However, the method [6] is applicable to neither high-dimensional systems nor constrained systems. An advantage of receding horizon control is that we can treat both constraints on the state variables and the control inputs. The objective of this study is to provide a generalized framework for designing a receding horizon controller for high-dimensional spatiotemporal dynamic systems with constraints.

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).

T. Hashimoto, Y. Yoshioka and T. Ohtsuka are with the Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan. Corresponding author's email: info@thashi.net

II. NOTATION AND SYSTEM MODEL

Let $\mathbb{R}$ denote the set of real numbers. Let $\mathbb{R}_+$ and $\mathbb{N}_+$ denote the sets of nonnegative real numbers and positive integers, respectively. For a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, the transpose of $\mathbf{A}$ is denoted by $\mathbf{A}'$. Let $n_x, n_z, n_d, n_b, n_c \in \mathbb{N}_+$ and $h \in \mathbb{R}_+\backslash\{0\}$ be positive constants. Furthermore, let $m_x, m_t, m_u, m_c, m_e \in \mathbb{N}_+$ be positive constants. Let $\mathbf{x} = [x_1, \cdots, x_{n_x}]' \in \mathbb{R}^{n_x}$ and $t \in \mathbb{R}_+$ denote a spatial vector and a temporal variable, respectively. Without loss of generality, we restrict our attention to the range $0 \le x_i \le h$ for all $i = 1, \cdots, n_x$. Let Ω and $\partial\Omega_i$ be the sets defined by $\Omega := \prod_{i=1}^{n_x} \{x_i | 0 \le x_i \le h\}$ and $\partial\Omega_i := \{x_i | x_i = 0, x_i = h\} \cap \Omega$, respectively. Let $\mathbf{z}(\mathbf{x}, t) = [z_1(\mathbf{x}, t), \cdots, z_{n_z}(\mathbf{x}, t)]' : \Omega \times \mathbb{R}_+ \to \mathbb{R}^{n_z}$ be a continuous vector-valued function with respect to $\mathbf{x}$ and t. For $i = 1, \cdots, n_x$ and $j = 1, \cdots, n_d$, we introduce the following notations:

$$\mathbf{z}_t(\mathbf{x}, t) := \frac{\partial \mathbf{z}(\mathbf{x}, t)}{\partial t} := \left[\frac{\partial z_1(\mathbf{x}, t)}{\partial t}, \cdots, \frac{\partial z_{n_z}(\mathbf{x}, t)}{\partial t}\right]',$$

$$\mathbf{z}_{x_i}(\mathbf{x}, t) := \frac{\partial \mathbf{z}(\mathbf{x}, t)}{\partial x_i} := \left[\frac{\partial z_1(\mathbf{x}, t)}{\partial x_i}, \cdots, \frac{\partial z_{n_z}(\mathbf{x}, t)}{\partial x_i}\right]',$$

$$\mathbf{z}_{x_i^j}(\mathbf{x}, t) := \frac{\partial^j \mathbf{z}(\mathbf{x}, t)}{\partial x_i^j} := \left[\frac{\partial^j z_1(\mathbf{x}, t)}{\partial x_i^j}, \cdots, \frac{\partial^j z_{n_z}(\mathbf{x}, t)}{\partial x_i^j}\right]',$$

$$z_{i\mathbf{x}}(\mathbf{x}, t) := \frac{\partial z_i(\mathbf{x}, t)}{\partial \mathbf{x}} := \left[\frac{\partial z_i(\mathbf{x}, t)}{\partial x_1}, \cdots, \frac{\partial z_i(\mathbf{x}, t)}{\partial x_{n_x}}\right],$$

$$\int_\Omega (\cdot) d\mathbf{x} := \int_0^h \cdots \int_0^h \int_0^h (\cdot) dx_1 dx_2 \cdots dx_{n_x},$$

$$\int_{\partial\Omega_i} (\cdot) d\mathbf{x} := \int_0^h \cdots \int_0^h \int_0^h \cdots dx_{i-1} dx_{i+1} \cdots dx_{n_x},$$

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

Let $\mathbb{D}$ be the set defined by $\mathbb{D} := \{\mathbf{z}(\mathbf{x}, t) : \Omega \times \mathbb{R}_+ \to \mathbb{R}^{n_z} |\ \mathbf{z}_t(\mathbf{x}, t)$ and $\mathbf{z}_{x_i^j}(\mathbf{x}, t)$ exist for all $i \in \{1, \cdots, n_x\}, j \in \{1, \cdots, n_d\}, \mathbf{x} \in \Omega, t \in \mathbb{R}_+\}$. Let $\mathbf{z}(\mathbf{x}, t) \in \mathbb{D}$ and $\mathbf{u}_i(\mathbf{x}, t) : \partial\Omega_i \times \mathbb{R}_+ \to \mathbb{R}^{n_z}$ $(i = 1, \cdots, n_x)$ be the state vector and the control input, respectively. Here, we consider the following spatiotemporal dynamic system written in a general setting:

$$\frac{\partial \mathbf{z}(\mathbf{x}, t)}{\partial t} = \mathbf{A}\left(\mathbf{z}, \mathbf{z}_{x_1}, \mathbf{z}_{x_1^2}, \cdots, \mathbf{z}_{x_i^j}, \cdots\right), \qquad \text{(1a)}$$

with boundary inputs

$$\frac{\partial \mathbf{z}(\mathbf{x},t)}{\partial x_i} = \mathbf{u}_i(\mathbf{x},t) \ \text{ for } \ \mathbf{x} \in \partial\Omega_i, \tag{1b}$$

boundary conditions

$$\mathbf{B}\left(\mathbf{z}, \mathbf{z}_{x_1}, \mathbf{z}_{x_1^2}, \cdots, \mathbf{z}_{x_i^j}, \cdots\right) = 0, \text{ for } \ \mathbf{x} \in \partial\Omega_i, \tag{1c}$$

and the initial condition $\mathbf{z}(\mathbf{x},0) = \mathbf{z}_0(\mathbf{x})$, where $\mathbf{A}$ and $\mathbf{B}$ are n_z-and n_b-dimensional vector-valued functions, respectively. Therein, $\mathbf{z}_0(\mathbf{x}) : \Omega \to \mathbb{R}^{n_z}$ denotes the initial state satisfying (1a)–(1d). Furthermore, equality constraints are imposed as

$$\mathbf{C}\left(\mathbf{u}_i, \mathbf{z}, \mathbf{z}_{x_1}, \mathbf{z}_{x_1^2}, \cdots, \mathbf{z}_{x_i^j}, \cdots\right) = 0, \tag{1d}$$

where $\mathbf{C}$ is an n_c-dimensional vector-valued function. It is well known that an inequality constraint can be converted into an equality constraint by introducing a slack variable [7]. Moreover, note that boundary conditions (1c) can be included in (1d).

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 (1a)–(1d) is unique and sufficiently smooth. Hereafter, we assume that $\mathbf{z}(\mathbf{x},t)$ is known at the present time t for all $\mathbf{x} \in \Omega$.

For example, mathematical models of the cooling process of a hot strip mill [8], magnetohydrodynamic turbulent flow [9], long waves in shallow water [10], and quantum dynamics [11] belong to a class of system (1a). Noting that hyperbolic PDEs containing $\partial^2 \mathbf{z}(\mathbf{x},t)/\partial t^2$ can be transformed into the form of (1a) by introducing an augmented state vector such as $\tilde{\mathbf{z}}(\mathbf{x},t) := [\mathbf{z}'(\mathbf{x},t), (\partial \mathbf{z}(\mathbf{x},t)/\partial t)']'$, we see that system models of the Timoshenko beam [12], Euler–Bernoulli beam [13], and transmission wave equation [14] can be reduced to a class of system (1a).

III. RECEDING HORIZON CONTROL

In this section, we consider the receding horizon control problem for a class of system (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 integrations by parts, which play an important role in this study.

The control input at each time t is determined so as to minimize the performance index given by

$$J = \int_\Omega \varphi[\mathbf{z}(\mathbf{x},t+T)]d\mathbf{x} + \int_t^{t+T}\int_\Omega L[\mathbf{z}(\mathbf{x},\tau), \mathbf{u}_i(\mathbf{x},\tau)]d\mathbf{x}d\tau, \tag{2}$$

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 (2) subject to (1) can be reduced to the minimization of the following performance index introduced using the costate $\boldsymbol{\lambda}(\mathbf{x},t) \in \mathbb{R}^{n_z}$ and the Lagrange multiplier $\boldsymbol{\mu}(\mathbf{x},t) \in \mathbb{R}^{n_c}$ associated with the equality constraints:

$$\bar{J} = \int_\Omega \varphi[\mathbf{z}(\mathbf{x},t+T)]d\mathbf{x} + \int_t^{t+T}\int_\Omega \left(H - \boldsymbol{\lambda}'\frac{\partial \mathbf{z}}{\partial \tau}\right) d\mathbf{x}d\tau. \tag{3}$$

Therein, $H \in \mathbb{R}$ denotes the Hamiltonian defined by

$$H\left(\mathbf{z}, \mathbf{z}_{x_i^j}, \mathbf{u}_i, \boldsymbol{\lambda}, \boldsymbol{\mu}\right) := L + \boldsymbol{\lambda}'\mathbf{A} + \boldsymbol{\mu}'\mathbf{C}. \tag{4}$$

Let $\delta\bar{J}$, $\delta\mathbf{z}$, $\delta\mathbf{z}_\tau$, $\delta\mathbf{z}_{x_i^j}$, $\delta\mathbf{u}$, $\delta\boldsymbol{\lambda}$, and $\delta\boldsymbol{\mu}$ denote the variations (infinitesimal changes) in $\bar{J}$, $\mathbf{z}$, $\mathbf{z}_\tau$, $\mathbf{z}_{x_i^j}$, $\mathbf{u}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$, respectively. It is important to note that we have the following integration by parts.

$$\begin{aligned}
&\int_t^{t+T}\int_\Omega -\boldsymbol{\lambda}'(\mathbf{x},\tau)\frac{\partial \delta\mathbf{z}(\mathbf{x},\tau)}{\partial \tau} d\mathbf{x}d\tau \\
&= \left[\int_\Omega -\boldsymbol{\lambda}'(\mathbf{x},\tau)\delta\mathbf{z}(\mathbf{x},\tau)d\mathbf{x}\right]_t^{t+T} \\
&\qquad + \int_t^{t+T}\int_\Omega \left(\frac{\partial \boldsymbol{\lambda}(\mathbf{x},\tau)}{\partial \tau}\right)' \delta\mathbf{z}(\mathbf{x},\tau)d\mathbf{x}d\tau \\
&= \int_\Omega -\boldsymbol{\lambda}'(\mathbf{x},t+T)\delta\mathbf{z}(\mathbf{x},t+T)d\mathbf{x} \\
&\qquad + \int_t^{t+T}\int_\Omega \left(\frac{\partial \boldsymbol{\lambda}(\mathbf{x},\tau)}{\partial \tau}\right)' \delta\mathbf{z}(\mathbf{x},\tau)d\mathbf{x}d\tau.
\end{aligned} \tag{5}$$

In the above, we set $\delta\mathbf{z}(\mathbf{x},t) = 0$ because $\mathbf{z}(\mathbf{x},\tau)$ is fixed at $\tau = t$ as the present state. The above integration by parts can be used to convert $\delta\mathbf{z}_\tau$ into $\delta\mathbf{z}$. It is also important to note that we can apply the following integrations by parts to the computation of $\delta\bar{J}$.

$$\begin{aligned}
&\int_\Omega \frac{\partial H}{\partial \mathbf{z}_{x_i}}\delta\mathbf{z}_{x_i}d\mathbf{x} = \\
&\int_{\partial\Omega_i}\left[\frac{\partial H}{\partial \mathbf{z}_{x_i}}\delta\mathbf{z}\right]_0^h d\mathbf{x} - \int_\Omega \frac{\partial}{\partial x_i}\left(\frac{\partial H}{\partial \mathbf{z}_{x_i}}\right)\delta\mathbf{z}d\mathbf{x},
\end{aligned} \tag{6}$$

$$\begin{aligned}
&\int_\Omega \frac{\partial H}{\partial \mathbf{z}_{x_i^2}}\delta\mathbf{z}_{x_i^2}d\mathbf{x} = \int_\Omega \frac{\partial^2}{\partial x_i^2}\left(\frac{\partial H}{\partial \mathbf{z}_{x_i^2}}\right)\delta\mathbf{z}d\mathbf{x} \\
&+ \int_{\partial\Omega_i}\left\{\left[\frac{\partial H}{\partial \mathbf{z}_{x_i^2}}\delta\mathbf{z}_{x_i}\right]_0^h - \left[\frac{\partial}{\partial x_i}\left(\frac{\partial H}{\partial \mathbf{z}_{x_i^2}}\right)\delta\mathbf{z}\right]_0^h\right\}d\mathbf{x},
\end{aligned} \tag{7}$$

$$\begin{aligned}
&\int_\Omega \frac{\partial H}{\partial \mathbf{z}_{x_i^j}}\delta\mathbf{z}_{x_i^j}d\mathbf{x} = (-1)^j\int_\Omega \left\{\frac{\partial^j}{\partial x_i^j}\left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}}\right)\delta\mathbf{z}\right\}d\mathbf{x} \\
&+ \int_{\partial\Omega_i}\sum_{k=1}^j\left\{(-1)^{j-k}\left[\frac{\partial^{j-k}}{\partial x_i^{j-k}}\left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}}\right)\delta\mathbf{z}_{x_i^{k-1}}\right]_0^h\right\}d\mathbf{x}.
\end{aligned} \tag{8}$$

From boundary condition (1b), we have

$$\delta\mathbf{z}_{x_i}(\mathbf{x},\tau) = \delta\mathbf{u}_i(\mathbf{x},\tau) \quad \text{for } \ \mathbf{x} \in \partial\Omega_i. \tag{9}$$

Substituting (9) into the term $\delta\mathbf{z}_{x_i}$ in (7) for $\mathbf{x} \in \partial\Omega_i$, we have the following:

$$\begin{aligned}
&\int_\Omega \frac{\partial H}{\partial \mathbf{z}_{x_i^2}}\delta\mathbf{z}_{x_i^2}d\mathbf{x} = \int_\Omega \frac{\partial^2}{\partial x_i^2}\left(\frac{\partial H}{\partial \mathbf{z}_{x_i^2}}\right)\delta\mathbf{z}d\mathbf{x} \\
&+ \int_{\partial\Omega_i}\left\{\left[\frac{\partial H}{\partial \mathbf{z}_{x_i^2}}\delta\mathbf{u}_i\right]_0^h - \left[\frac{\partial}{\partial x_i}\left(\frac{\partial H}{\partial \mathbf{z}_{x_i^2}}\right)\delta\mathbf{z}\right]_0^h\right\}d\mathbf{x}.
\end{aligned} \tag{10}$$

For $j \geq 3$, substituting (9) into the term $\delta \mathbf{z}_{x_i}$ in (8) for $\mathbf{x} \in \partial\Omega_i$ yields

$$\int_\Omega \frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \delta \mathbf{z}_{x_i^j} d\mathbf{x} = (-1)^j \int_\Omega \left\{ \frac{\partial^j}{\partial x_i^j} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) \delta \mathbf{z} \right\} d\mathbf{x}$$
$$+ \int_{\partial\Omega_i} \left((-1)^{j-1} \left[\frac{\partial^{j-1}}{\partial x_i^{j-1}} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) \delta \mathbf{z} \right]_0^h \right.$$
$$+ (-1)^{j-2} \left[\frac{\partial^{j-2}}{\partial x_i^{j-2}} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) \delta \mathbf{u}_i \right]_0^h$$
$$\left. + \sum_{k=3}^{j} \left\{ (-1)^{j-k} \left[\frac{\partial^{j-k}}{\partial x_i^{j-k}} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) \delta \mathbf{z}_{x_i^{k-1}} \right]_0^h \right\} \right) d\mathbf{x}. \tag{11}$$

Note that the variations $\delta \mathbf{z}_{x_i^j}$ for $\mathbf{x} \in \Omega$, $j \in \{1, \cdots, n_d\}$ can be resolved into $\delta \mathbf{z}$ for $\mathbf{x} \in \Omega$, $\delta \mathbf{u}_i$ for $\mathbf{x} \in \partial\Omega_i$, and $\delta \mathbf{z}_{x_i^j}$ for $\mathbf{x} \in \partial\Omega_i$, $j \in \{2, \cdots, n_d\}$. First, using the integration by parts in (5), we obtain the variation in $\bar{J}$ as follows:

$$\delta \bar{J} = \int_\Omega \left\{ \frac{\partial \varphi[\mathbf{z}(\mathbf{x}, t+T)]}{\partial \mathbf{z}(\mathbf{x}, t+T)} - \boldsymbol{\lambda}'(\mathbf{x}, t+T) \right\} \delta \mathbf{z}(\mathbf{x}, t+T) d\mathbf{x}$$
$$+ \int_t^{t+T} \int_\Omega \left\{ \delta H[\mathbf{x}, \tau] - \delta \boldsymbol{\lambda}'(\mathbf{x}, \tau) \frac{\partial \mathbf{z}(\mathbf{x}, \tau)}{\partial \tau} \right.$$
$$\left. + \left(\frac{\partial \boldsymbol{\lambda}(\mathbf{x}, \tau)}{\partial \tau} \right)' \delta \mathbf{z}(\mathbf{x}, \tau) \right\} d\mathbf{x} d\tau, \tag{12}$$

where δH is given by

$$\delta H \left(\mathbf{z}, \mathbf{z}_{x_i^j}, \mathbf{u}_i, \boldsymbol{\lambda}, \boldsymbol{\mu} \right) = \left(\frac{\partial H}{\partial \mathbf{z}} \right) \delta \mathbf{z} + \sum_{i=1}^{n_x} \sum_{j=1}^{n_d} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) \delta \mathbf{z}_{x_i^j}$$
$$+ \sum_{i=1}^{n_x} \left(\frac{\partial H}{\partial \mathbf{u}_i} \right) \delta \mathbf{u}_i + \left(\frac{\partial H}{\partial \boldsymbol{\lambda}} \right) \delta \boldsymbol{\lambda} + \left(\frac{\partial H}{\partial \boldsymbol{\mu}} \right) \delta \boldsymbol{\mu}. \tag{13}$$

Next, we apply the integrations by parts in (6), (10), and (11) to the computation of δH in (13).

$$\int_\Omega \delta H \left(\mathbf{z}, \mathbf{z}_{x_i^j}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\mu} \right) d\mathbf{x} = \int_\Omega \left[\delta \boldsymbol{\lambda}' \mathbf{A} + \delta \boldsymbol{\mu}' \mathbf{C} \right.$$
$$\left. \left\{ \frac{\partial H}{\partial \mathbf{z}} + \sum_{i=1}^{n_x} \sum_{j=1}^{n_d} (-1)^j \frac{\partial^j}{\partial x_i^j} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) \right\} \delta \mathbf{z} \right] d\mathbf{x}$$
$$+ \sum_{i=1}^{n_x} \int_{\partial\Omega_i} \left\{ \sum_{j=1}^{n_d} \left[(-1)^{j-1} \frac{\partial^{j-1}}{\partial x_i^{j-1}} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) \delta \mathbf{z} \right]_0^h \right.$$
$$+ \sum_{j=3}^{n_d} \sum_{k=3}^{j} \left[(-1)^{j-k} \frac{\partial^{j-k}}{\partial x_i^{j-k}} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) \delta \mathbf{z}_{x_i^{k-1}} \right]_0^h$$
$$\left. + \sum_{j=2}^{n_d} \left[(-1)^{j-2} \frac{\partial^{j-2}}{\partial x_i^{j-2}} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) \delta \mathbf{u}_i \right]_0^h \right\} d\mathbf{x}$$
$$+ \sum_{i=1}^{n_x} \int_\Omega \left(\frac{\partial H}{\partial \mathbf{u}_i} \right) \delta \mathbf{u}_i d\mathbf{x}. \tag{14}$$

For the last term in (14), we have the following:

$$\int_\Omega \left(\frac{\partial H}{\partial \mathbf{u}_i} \right) \delta \mathbf{u}_i d\mathbf{x} = \int_{\partial\Omega_i} \left\{ \sum_{x_i \in \{0,h\}} \left(\frac{\partial H}{\partial \mathbf{u}_i} \right) \delta \mathbf{u}_i \right\} d\mathbf{x}. \tag{15}$$

Taking (12)–(15) into consideration, we obtain $\delta \bar{J}$ as follows:

$$\delta \bar{J} = \int_\Omega \left\{ \frac{\partial \varphi[\mathbf{z}]}{\partial \mathbf{z}} - \boldsymbol{\lambda}'(\mathbf{x}, t+T) \right\} \delta \mathbf{z}(\mathbf{x}, t+T) d\mathbf{x}$$
$$+ \int_t^{t+T} \left[\int_\Omega \left[\delta \boldsymbol{\lambda}' \left(\mathbf{A} - \frac{\partial \mathbf{z}}{\partial \tau} \right) + \delta \boldsymbol{\mu}' \boldsymbol{C} \right. \right.$$
$$\left. \left\{ \left(\frac{\partial \boldsymbol{\lambda}}{\partial \tau} \right)' + \frac{\partial H}{\partial \mathbf{z}} + \sum_{i=1}^{n_x} \sum_{j=1}^{n_d} (-1)^j \frac{\partial^j}{\partial x_i^j} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) \right\} \delta \mathbf{z} \right] d\mathbf{x}$$
$$+ \sum_{i=1}^{n_x} \int_{\partial\Omega_i} \sum_{x_i \in \{0,h\}} \left[\left\{ \left(\frac{\partial H}{\partial \mathbf{u}_i} \right) \right. \right.$$
$$\left. + \sum_{j=2}^{n_d} s(x_i) (-1)^{j-2} \frac{\partial^{j-2}}{\partial x_i^{j-2}} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) \right\} \delta \mathbf{u}_i$$
$$+ s(x_i) \sum_{j=3}^{n_d} \left\{ \sum_{k=0}^{n_d - j} (-1)^k \frac{\partial^k}{\partial x_i^k} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^{j+k}}} \right) \right\} \delta \mathbf{z}_{x_i^{j-1}}$$
$$\left. \left. + \; s(x_i) \left\{ \sum_{j=1}^{n_d} (-1)^{j-1} \frac{\partial^{j-1}}{\partial x_i^{j-1}} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) \right\} \delta \mathbf{z} \right] d\mathbf{x} \right] d\tau. \tag{16}$$

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 $\mathbf{x} \in \Omega$, we have

$$\frac{\partial \mathbf{z}(\mathbf{x}, \tau)}{\partial \tau} = \mathbf{A} \left(\mathbf{z}, \mathbf{z}_{x_1}, \mathbf{z}_{x_1^2}, \cdots, \mathbf{z}_{x_i^j}, \cdots \right), \tag{17a}$$

$$\boldsymbol{\lambda}(\mathbf{x}, t+T) = \left\{ \frac{\partial \varphi[\mathbf{z}(\mathbf{x}, t+T)]}{\partial \mathbf{z}(\mathbf{x}, t+T)} \right\}', \tag{17b}$$

$$\left(\frac{\partial \boldsymbol{\lambda}}{\partial \tau} \right) = \left\{ -\frac{\partial H}{\partial \mathbf{z}} - \sum_{i=1}^{n_x} \sum_{j=1}^{n_d} (-1)^j \frac{\partial^j}{\partial x_i^j} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) \right\}', \tag{17c}$$

$$\mathbf{C} \left(\mathbf{u}_i, \mathbf{z}, \mathbf{z}_{x_1}, \mathbf{z}_{x_1^2}, \cdots, \mathbf{z}_{x_i^j}, \cdots \right) = 0, \tag{17d}$$

and, for $i = 1, \cdots, n_x$ and $x_i \in \partial\Omega_i$, we have

$$\left(\frac{\partial H}{\partial \mathbf{u}_i} \right) + \sum_{j=2}^{n_d} s(x_i) (-1)^{j-2} \frac{\partial^{j-2}}{\partial x_i^{j-2}} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) = 0, \tag{17e}$$

$$\sum_{j=1}^{n_d} (-1)^{j-1} \frac{\partial^{j-1}}{\partial x_i^{j-1}} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^j}} \right) = 0, \tag{17f}$$

and, for $i = 1, \cdots, n_x$, $x_i \in \partial\Omega_i$, and $j = 3, \cdots, n_x$, we have

$$\sum_{k=0}^{n_d - j} (-1)^k \frac{\partial^k}{\partial x_i^k} \left(\frac{\partial H}{\partial \mathbf{z}_{x_i^{j+k}}} \right) = 0. \tag{17g}$$

Conditions (17a)–(17g) are called the stationary conditions, which must be satisfied for the performance index (3) to be minimized. A well-known difficulty in solving nonlinear optimal control problems is that the obtained stationary conditions cannot be solved analytically in general.

IV. NUMERICAL SOLUTION

Although we have analytically derived the exact stationary conditions in Sec. III, we need a numerical algorithm for solving the stationary conditions. In the following, we provide a framework in which a fast on-line algorithm called C/GMRES [5] is applicable for solving the receding horizon control problem in nonlinear PDEs.

Here, note that the stationary conditions contain two time-evolutionary equations with respect to $\mathbf{z}$ and $\boldsymbol{\lambda}$. The others are algebraic equations, and (17f)–(17g) are considered as the boundary conditions for the time-evolutionary equation of $\boldsymbol{\lambda}$. Let $\mathbf{U}$ be defined by $\mathbf{U} = [\mathbf{u}_1', \cdots, \mathbf{u}_{n_x}', \boldsymbol{\mu}']'$. For a given initial solution $\mathbf{U}(\mathbf{x}, \tau)$, $\tau \in [t,\ t+T]$ and the present state $\mathbf{z}(\mathbf{x}, t)$, we first determine $\mathbf{z}(\mathbf{x}, \tau)$ for $\tau \in [t,\ t+T]$ by numerically solving (17a) with boundary conditions (1b)–(1c) from $\tau = t$ to $\tau = t + T$. Then, the terminal costate $\boldsymbol{\lambda}(\mathbf{x}, t+T)$ is determined from the obtained terminal state $\mathbf{z}(\mathbf{x}, t+T)$ by (17b). Consequently, $\boldsymbol{\lambda}(\mathbf{x}, \tau)$ for $\tau \in [t,\ t+T]$ is also determined by numerically solving (17c) with boundary conditions (17f)–(17g) from $\tau = t + T$ to $\tau = t$. Figure 1 shows that the procedure for solving the time-evolutionary equation of $\mathbf{z}$ is forward, whereas the one for solving the time-evolutionary equation of $\boldsymbol{\lambda}$ is backward.

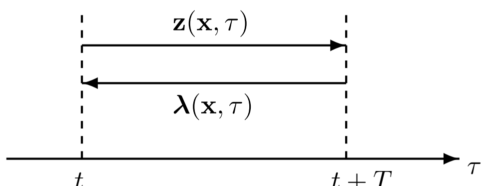

Fig. 1. Procedure used for obtaining numerical solutions.

To solve stationary conditions (17a)–(17g) using a numerical algorithm, we must discretize equations (17a)–(17g) into finite difference equations. Let $\mathbf{x} \in \Omega$ be divided into m_x grid points, and let $\hat{\mathbf{x}} := [\hat{x}_1, \cdots, \hat{x}_{m_x}]' \in \mathbb{R}^{m_x}$ 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 let $\hat{\boldsymbol{\tau}} := [\hat{\tau}_1, \cdots, \hat{\tau}_{m_t}]' \in \mathbb{R}^{m_t}$ denote the discretized temporal vector. Note that $\hat{\tau}_1$ is identical to the present time t. Let the set $\{\partial \hat{x}_{i,1}, \cdots, \partial \hat{x}_{i,m_u}\}$ be given by $\{\hat{x}_1, \cdots, \hat{x}_{m_x}\} \cap \partial\Omega_i$. Let $\partial \hat{\mathbf{x}}_i \in \mathbb{R}^{m_u}$ be defined by $\partial \hat{\mathbf{x}}_i := [\partial \hat{x}_{i,1}, \cdots, \partial \hat{x}_{i,m_u}]'$. Let $\hat{\mathbf{u}}(\partial \hat{\mathbf{x}}, \hat{\boldsymbol{\tau}}) := [\hat{\mathbf{u}}_1'(\partial \hat{\mathbf{x}}_1, \hat{\boldsymbol{\tau}}), \cdots, \hat{\mathbf{u}}_{n_x}'(\partial \hat{\mathbf{x}}_{n_x}, \hat{\boldsymbol{\tau}})]'$ denote the discretized control input. Let $\hat{\mathbf{z}}(\hat{\mathbf{x}}, \hat{\boldsymbol{\tau}})$, $\hat{\boldsymbol{\lambda}}(\hat{\mathbf{x}}, \hat{\boldsymbol{\tau}})$, and $\hat{\boldsymbol{\mu}}(\hat{\mathbf{x}}, \hat{\boldsymbol{\tau}})$ denote the discretized state, costate, and Lagrange multiplier, respectively. For notational simplicity, let $\hat{\mathbf{z}}(\hat{\mathbf{x}}, \hat{\tau}_k)$ be denoted by $\hat{\mathbf{z}}_k$ for $k = 1, \cdots, m_t$. Note that $\hat{\mathbf{z}}_1$ is identical to the present known state $\hat{\mathbf{z}}(\hat{\mathbf{x}}, t)$. For other variables, we adopt such notation without explanation. As a result of the finite difference approximation, we obtain the discretized stationary conditions over the horizon $(k = 1, \cdots, m_t)$ as follows:

$$\hat{\mathbf{z}}_{k+1} = \hat{\mathbf{A}}\left(\hat{\mathbf{z}}_k, \hat{\mathbf{u}}_k\right), \tag{18a}$$

$$\hat{\boldsymbol{\lambda}}_{m_t} = \hat{\boldsymbol{\Phi}}\left(\hat{\mathbf{z}}_{m_t}\right), \tag{18b}$$

$$\hat{\boldsymbol{\lambda}}_k = \hat{\mathbf{D}}\left(\hat{\boldsymbol{\lambda}}_{k+1}, \hat{\mathbf{u}}_{k+1}, \hat{\boldsymbol{\mu}}_{k+1}\right), \tag{18c}$$

$$\hat{\mathbf{C}}_k\left(\hat{\mathbf{u}}_k, \hat{\mathbf{z}}_k\right) = 0, \tag{18d}$$

$$\hat{\mathbf{E}}_k\left(\hat{\mathbf{u}}_k, \hat{\mathbf{z}}_k, \hat{\boldsymbol{\lambda}}_k, \hat{\boldsymbol{\mu}}_k\right) = 0. \tag{18e}$$

where $\hat{\mathbf{A}}$, $\hat{\boldsymbol{\Phi}}$, and $\hat{\mathbf{D}}$ are m_x-dimensional vector-valued functions; $\hat{\mathbf{C}}$ is an m_c-dimensional vector-valued function; and $\hat{\mathbf{E}}$ is an m_e-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 (18) without referring to any specific discretization method. The time-evolutionary equations of $\mathbf{z}$ and $\boldsymbol{\lambda}$ are discretized into forward difference equation (18a) and backward difference equation (18c), respectively. Note that boundary conditions (1b)–(1c) are also discretized and employed in (18a). Moreover, the equations obtained by discretizing (17c), (17f), and (17g) are unified into (18c). The remaining stationary conditions (17b), (17d), and (17e) are also discretized and described in general forms as (18b), (18d), and (18e), respectively.

For the present time t, let unknown parameters $\hat{\mathbf{u}}_k$ and $\hat{\boldsymbol{\mu}}_k$ for $k = 1, \cdots, m_t$ be combined into the vector defined by $\hat{\mathbf{U}}(t) := [\hat{\mathbf{u}}_1', \cdots, \hat{\mathbf{u}}_{m_t}', \hat{\boldsymbol{\mu}}_1', \cdots, \hat{\boldsymbol{\mu}}_{m_t}']'$. For the present state $\hat{\mathbf{z}}_1(t)$ and a given initial solution $\hat{\mathbf{U}}(t)$, $\hat{\mathbf{z}}_k(t)$ for $k = 1, \cdots, m_t$ is calculated recursively from $k = 1$ to $k = m_t$ by (18a). Next, the terminal costate $\hat{\boldsymbol{\lambda}}_{m_t}(t)$ is determined from the terminal state $\hat{\mathbf{z}}_{m_t}(t)$ by (18b). Consequently, $\hat{\boldsymbol{\lambda}}_k(t)$ for $k = 1, \cdots, m_t$ is also calculated recursively from $k = m_t$ to $k = 1$ by (18c). Since $\hat{\mathbf{z}}_k(t)$ and $\hat{\boldsymbol{\lambda}}_k(t)$ are determined by $\hat{\mathbf{z}}_1(t)$ and $\hat{\mathbf{U}}(t)$ through (18a)–(18c), equations (18d)–(18e) can be regarded as the single equation

$$\mathbf{F}\left(\hat{\mathbf{U}}(t), \hat{\mathbf{z}}_1(t), t\right) := \left[\hat{\mathbf{C}}_1', \cdots, \hat{\mathbf{C}}_{m_t}', \hat{\mathbf{E}}_1', \cdots \hat{\mathbf{E}}_{m_t}'\right]' \in \mathbb{R}^{m_f}, \tag{19}$$

where $m_f := m_c m_t + m_e m_t$. Since $\hat{\mathbf{z}}_k(t)$ and $\hat{\boldsymbol{\lambda}}_k(t)$ are uniquely determined through (18a)–(18c) for the given $\hat{\mathbf{z}}_1(t)$ and $\hat{\mathbf{U}}(t)$, $\mathbf{z}_k(t)$ and $\hat{\boldsymbol{\lambda}}_k(t)$ depend on $\hat{\mathbf{z}}_1(t)$ and $\hat{\mathbf{U}}(t)$. Hence, it is reasonable to consider the arguments of $\mathbf{F}$ as $\hat{\mathbf{U}}(t), \hat{\mathbf{z}}_1(t), t$.

For given $\hat{\mathbf{z}}_1(t)$ and $\hat{\mathbf{U}}(t)$, $\mathbf{F}$ is not necessarily equal to zero. $\|\mathbf{F}\|$ is used to evaluate the optimality performance. If $\|\mathbf{F}\| = 0$ is satisfied for the given $\hat{\mathbf{z}}_1(t)$ and $\hat{\mathbf{U}}(t)$, then the stationary conditions are satisfied. Several algorithms have been developed such that $\|\mathbf{F}\|$ can be decreased by suitably updating $\hat{\mathbf{U}}(t)$.

A conventional way of updating $\hat{\mathbf{U}}(t)$ is to replace $\hat{\mathbf{U}}(t)$ with $\hat{\mathbf{U}}(t) + \alpha\mathbf{s}$, which is known as the steepest descent method, where $\mathbf{s}$ is the steepest descent direction and α is the step length satisfying the Armijo condition [15]. For Newton's method, $\mathbf{s}$ is given by the Hessian instead of the gradient. However, these methods are computationally expensive, and it was shown that the C/GMRES algorithm [5] is not only faster but also more numerically robust than such conventional algorithms. Instead of solving $\mathbf{F}(\hat{\mathbf{U}}(t), \hat{\mathbf{z}}_1(t), t) = 0$ itself at each time by an iterative method such as the steepest descent method or Newton's method, we find the derivative of $\hat{\mathbf{U}}(t)$ with respect to time such that $\mathbf{F}(\hat{\mathbf{U}}(t), \hat{\mathbf{z}}_1(t), t) = 0$ is satisfied identically. Namely, we determine $\dot{\hat{\mathbf{U}}}(t)$ such that

$$\dot{\mathbf{F}}(\hat{\mathbf{U}}(t), \hat{\mathbf{z}}_1(t), t) = -\xi \mathbf{F}(\hat{\mathbf{U}}(t), \hat{\mathbf{z}}_1(t), t) \tag{20}$$

is satisfied, where ξ is a positive constant introduced to stabilize $\mathbf{F} = 0$. By total differentiation, we have

$$\frac{\partial \mathbf{F}}{\partial \hat{\mathbf{U}}(t)} \dot{\hat{\mathbf{U}}}(t) = -\xi \mathbf{F} - \frac{\partial \mathbf{F}}{\partial \hat{\mathbf{z}}_1} \dot{\hat{\mathbf{z}}}_\mathbf{1} - \frac{\partial \mathbf{F}}{\partial \mathbf{t}}. \tag{21}$$

This can be regarded as a linear algebraic equation with the coefficient matrix $(\partial \mathbf{F}/\partial \hat{\mathbf{U}}(t))$, which can be used to determine $\dot{\hat{\mathbf{U}}}$ for the given $\hat{\mathbf{U}}$, $\hat{\mathbf{z}}_1$, $\dot{\hat{\mathbf{z}}}_\mathbf{1}$, and t. Then, if the Jacobian $(\partial \mathbf{F}/\partial \hat{\mathbf{U}})$ is nonsingular, we obtain the following differential equation for $\hat{\mathbf{U}}(t)$:

$$\dot{\hat{\mathbf{U}}}(t) = \left(\frac{\partial \mathbf{F}}{\partial \hat{\mathbf{U}}(t)} \right)^{-1} \left(-\xi \mathbf{F} - \frac{\partial \mathbf{F}}{\partial \hat{\mathbf{z}}_1} \dot{\hat{\mathbf{z}}}_\mathbf{1}(\mathbf{t}) - \frac{\partial \mathbf{F}}{\partial t} \right). \tag{22}$$

We can update the solution $\hat{\mathbf{U}}(t)$ of $\mathbf{F}(\hat{\mathbf{U}}(t), \hat{\mathbf{z}}_1(t), t) = 0$ without using an iterative optimization method by integrating (22) in real time as, for example, $\hat{\mathbf{U}}(t + \Delta t) = \hat{\mathbf{U}}(t) + \dot{\hat{\mathbf{U}}}(t)\Delta t$. More detailed information about the implementation of C/GMRES is provided in [5].

Recently, we have developed a more efficient algorithm than C/GMRES, which is called the contraction mapping method [6]. More precisely, we have proposed the algorithm for solving $\mathbf{F}(\mathbf{U}, \hat{\mathbf{z}}_1, t) = 0$ under the assumption that $\mathbf{F}(\hat{\mathbf{U}}, \hat{\mathbf{z}}_1, t)$ satisfies a particular structural condition with respect to $\hat{\mathbf{U}}$. Compared with the C/GMRES method [5], the contraction mapping method [6] has the disadvantage of limited applicability but the advantage of a smaller computational burden. In the following, a brief description of the contraction mapping method is provided. Hereafter, we assume that $\mathbf{F}$ is given by

$$\mathbf{F}(\hat{\mathbf{U}}(t), \hat{\mathbf{z}}_1(t), t) = \mathbf{Q}\hat{\mathbf{U}}(t) - \mathbf{R}(\hat{\mathbf{U}}(t), \hat{\mathbf{z}}_1(t), t), \tag{23}$$

where $\mathbf{Q} \in \mathbb{R}^{m_f \times m_f}$ is a nonsingular constant matrix and $\mathbf{R} \in \mathbb{R}^{m_f}$ is a vector-valued function satisfying the following Lipschitz continuous conditions: for v_1, v_2, v_3, $v_4 \in \mathbb{R}_+$,

$$\begin{aligned}
&\left\| \mathbf{R}\left(\hat{\mathbf{U}}_{(t+\Delta t)}, \hat{\mathbf{z}}_{1(t+\Delta t)}, t + \Delta t\right) - \mathbf{R}\left(\hat{\mathbf{U}}_{(t)}, \hat{\mathbf{z}}_{1(t)}, t\right) \right\| \\
&\leq v_1 \left\| \hat{\mathbf{U}}_{(t+\Delta t)} - \hat{\mathbf{U}}_{(t)} \right\| + v_2 \left\| \hat{\mathbf{z}}_{1(t+\Delta t)} - \hat{\mathbf{z}}_{1(t)} \right\| + v_3 \Delta t, \\
&\left\| \hat{\mathbf{z}}_{1(t+\Delta t)} - \hat{\mathbf{z}}_{1(t)} \right\| \leq v_4 \Delta t.
\end{aligned}$$

Let $\mathbf{P} \in \mathbb{R}^{m_f}$ be defined by

$$\mathbf{P}(\hat{\mathbf{U}}, \hat{\mathbf{z}}_1, t) = \mathbf{Q}^{-1}\mathbf{R}(\hat{\mathbf{U}}, \hat{\mathbf{z}}_1, t). \tag{24}$$

Here, we adopt the following notations:

$$\mathbf{P} \circ \mathbf{P}(\hat{\mathbf{U}}, \hat{\mathbf{z}}_1, t) = \mathbf{P}(\mathbf{P}(\hat{\mathbf{U}}, \hat{\mathbf{z}}_1, t), \hat{\mathbf{z}}_1, t), \tag{25a}$$

$$\mathbf{P}^k(\hat{\mathbf{U}}, \hat{\mathbf{z}}_1, t) = \underbrace{\mathbf{P} \circ \cdots \circ \mathbf{P}}_{k}(\hat{\mathbf{U}}, \hat{\mathbf{z}}_1, t). \tag{25b}$$

Suppose that $\hat{\mathbf{U}}$ is updated as

$$\hat{\mathbf{U}}(t) = \mathbf{P}^k(\hat{\mathbf{U}}(t - \Delta t), \hat{\mathbf{z}}_1(t), t), \tag{26}$$

where $t = \Delta t, 2\Delta t, \cdots, j\Delta t$ for $j \in \mathbb{N}_+$ and $k \in \mathbb{N}_+$ is a design parameter. Then, we can state the following theorem.

Theorem 1 ([6]): If

$$v_1 \left\| \mathbf{Q}^{-1} \right\| < 1 \tag{27}$$

is satisfied, then $\|\mathbf{F}(t)\|$ is ultimately bounded as follows:

$$\|\mathbf{F}(t - \Delta t)\| \leq \delta \Rightarrow \|\mathbf{F}(t)\| \leq \delta, \tag{28a}$$

$$\|\mathbf{F}(t - \Delta t)\| > \delta \Rightarrow \|\mathbf{F}(t)\| < \|\mathbf{F}(t - \Delta t)\|, \tag{28b}$$

where δ is given by

$$\delta = \frac{\left(v_1 \left\| \mathbf{Q}^{-1} \right\|\right)^k}{1 - \left(v_1 \left\| \mathbf{Q}^{-1} \right\|\right)^k} \left(v_2 v_4 + v_3\right) \Delta t. \tag{29}$$

Note that $\|\mathbf{F}\|$ decreases monotonically whenever $\|\mathbf{F}\| > \delta$. It is worth noting that the upper bound δ of $\|\mathbf{F}\|$ converges to zero as k increases to infinity. From this viewpoint, the contraction mapping method provides a satisfactory trade-off between computational burden and error performance through the selection of design parameter k.

V. ILLUSTRATIVE EXAMPLE

In this section, we provide an illustrative example to verify the effectiveness of the proposed method. For two-dimensional square domain $\mathbf{x} \in \Omega := [0\ \ 0.1] \times [0\ \ 0.1]$, we consider an incompressible flow of thermal fluid dynamics that can be described by the Boussinesq equations consisting of the momentum, continuity, and energy equations:

$$\frac{\partial \mathbf{v}}{\partial t} = -\frac{1}{\rho}\nabla P - (\mathbf{v} \cdot \nabla)\mathbf{v} + \nu \nabla^2 \mathbf{v} + \mathbf{g}\beta(\theta - \theta_0), \tag{30a}$$

$$\nabla \cdot \mathbf{v} = 0, \tag{30b}$$

$$\frac{\partial \theta}{\partial t} = -(\mathbf{v} \cdot \nabla)\theta + \alpha \nabla^2 \theta. \tag{30c}$$

Therein, $\mathbf{v}(\mathbf{x}, t) \in \mathbb{R}^2$ is the velocity [m/s], $\theta(\mathbf{x}, t) \in \mathbb{R}$ is the temperature [K], and $P(\mathbf{x}, t) \in \mathbb{R}$ is the pressure [N/m^2], and they are considered as the state vector $\mathbf{z} := [\mathbf{v}', \theta, P]'$. The other notations in (30) are all constant parameters as follows: $\rho \in \mathbb{R}$ is the density [kg/m^3], $\nu \in \mathbb{R}$ is the kinematic viscosity [m^2/s], $\mathbf{g} \in \mathbb{R}^2$ is the gravity acceleration [m/s^2], θ_0 is a reference temperature [K], α is the thermal diffusivity coefficient [m^2/s], and β is the thermal expansion coefficient [1/K]. Here, we consider the following boundary conditions:

$$\mathbf{v} = 0,\ \theta_{x_1} = u_1,\ P_{x_1} = 0,\ \ \text{for } \mathbf{x} \in \partial\Omega_1, \tag{31a}$$

$$\mathbf{v} = 0,\ \theta_{x_2} = u_2,\ P_{x_2} = 0,\ \ \text{for } \mathbf{x} \in \partial\Omega_2, \tag{31b}$$

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

Here, we consider the temperature control problem of the air flow that is governed by (30). 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 system parameters as follows: $\rho = 1.25$, $\nu = 1.38 \times 10^{-5}$, $\mathbf{g} = [0, 9.8]'$, $\theta_0 = 300$, $\alpha = 1.91 \times 10^{-5}$, and $\beta = 3.33 \times 10^{-3}$. The desired temperature $\theta_f(\mathbf{x})$ is set as $\theta_f(\mathbf{x}) = 310$ for all $x \in \Omega$. Here, we introduce the following performance index:

$$\varphi = \frac{1}{2} w_1 \left\{ \theta(\mathbf{x}, t+T) - \theta_f \right\}^2, \tag{32a}$$

$$L = \frac{1}{2} \left[w_2 \left\{ \theta(\mathbf{x}, \tau) - \theta_f \right\}^2 + w_3 \sum_{i=1}^{2} u_i^2(\mathbf{x}, \tau) \right]. \tag{32b}$$

In this example, we can solve the optimization problem by applying the contraction mapping method [6], in which we adopt a staggered grid and the simplified marker and cell method [18] for numerical computation of the time-evolutionary equations of $\mathbf{z}$ and $\boldsymbol{\lambda}$.

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: $\Delta t = 0.02$, $m_x = 30 \times 30$, $m_t = 10$, $[w_1, w_2, w_3] = [1, 10^4, 10^{-5}]$, and $k = 1$. Owing to the initialization of the optimal solution $\mathbf{U}(0)$, the length of the horizon is selected such that $T(0) = 0$ and $T(t) \to 0.1$ as $t \to \infty$, that is, $T = 0.1(1 - e^{-0.5t})$.

Figures 2–5 and figures 6–8 show the time response of temperature θ and flow velocity $\mathbf{v}$, respectively, controlled by the receding horizon control via the contraction mapping method. The figures reveal the effectiveness of the proposed method. Figures 9 and 10 show the time response of control inputs u_1 at $x_1 = 0.1$ and u_2 at $x_2 = 0$, respectively. The time response of control inputs u_1 at $x_1 = 0$ and u_2 at $x_2 = 0.1$ are omitted for shortage of space. Simulation is performed on a personal computer (CPU: Core 2 Duo 2.80 [GHz], Memory: 2.96 [GB], OS: Windows XP, Software: Matlab). The average computational time per update (one control cycle) is 1.6 [s] by the contraction mapping method.

VI. CONCLUSION

In this study, we provided a generalized framework of designing a receding horizon controller for high-dimensional spatiotemporal dynamic systems with constraints. The method proposed here 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. 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.

REFERENCES

[1] J. L. Lions and S. K. Mitter, *Optimal Control of Systems Governed by Partial Differential Equations*, Springer, 1971.

[2] R. F. Curtain and H. J. Zwart, *An Introduction to Infinite-Dimensional Linear Systems Theory*, Texts in Applied Mathematics, 21, Springer-Verlag, 1995.

[3] M. Krstic and A. Smyshlyaev, *Boundary Control of PDEs*, Advances in Design and Control, 16, SIAM Philadelphia, 2008.

[4] A. Bemporad, M. Morari, V. Dua and E.N. Pistikopoulos, The Explicit Linear Quadratic Regulator for Constrained Systems, *Automatica*, Vol. 38, 2002, pp. 3-20.

[5] T. Ohtsuka, A Continuation/GMRES Method for Fast Computation of Nonlinear Receding Horizon Control, *Automatica*, Vol. 40, 2004, pp. 563-574.

[6] T. Hashimoto, Y. Yoshioka and T. Ohtsuka, Receding Horizon Control for Nonlinear Parabolic Partial Differential Equations with Boundary Control Inputs, *Proceedings of the 49th IEEE Conference on Decision and Control*, pp. 6920-6925, 2010.

[7] A. Bryson and Y. C. Ho, *Applied Optimal Control*, Hemisphere Publishing, Washington, 1975.

[8] R. K. Kumar, S. K. Sinha and A. K. Lahiri, An On-Line Parallel Controller for the Runout Table of Hot Strip Mills, *IEEE Transactions on Control Systems Technology*, vol. 9, 2001, pp. 821-830.

[9] U. Muller and L. Buhler, *Magnetofluid Dynamics in Channels and Containers*, Springer, Berlin, 2001.

[10] A. Balogh and M. Krstic, Boundary Control of the Korteweg-de Vries-Burgers Equation: Further Results on Stabilization and Numerical Demonstration, *IEEE Transactions on Automatic Control*, Vol. 45, 2000, pp. 1739-1745.

[11] M. Krstic, B.-Z. Guo and A. Smyshlyaev, Boundary Controllers and Observers for the Linearized Schrodinger Equation, *SIAM Journal of Control and Optimization*, Vol. 49, 2011, pp. 1479-1497.

[12] A. Macchelli and C. Melchiorri, Modeling and Control of the Timoshenko Beam. The Distributed Port Hamiltonian Approach, *SIAM Journal of Control and Optimization*, Vol. 43, 2004, pp. 743-767.

[13] A. Smyshlyaev, B.-Z. Guo and M. Krstic, Arbitrary Decay Rate for Euler-Bernoulli Beam by Backstepping Boundary Feedback *IEEE Transactions on Automatic Control*, Vol. 54, 2009, pp. 1134-1140.

[14] W. Liu Stabilization and Controllability for the Transmission Wave Equation, *IEEE Transactions on Automatic Control*, Vol. 46, 2001, pp. 1900-1907.

[15] J. Nocedal and S. J. Wright, Numerical optimization, *Springer Series in Operation Research and Financial Engineering*, Springer, 2006, Chap. 3.

[16] S. L. Richter and R. A. DeCarlo, Continuation methods: theory and applications, *IEEE Transactions on Automatic Control*, Vol. 28, 1983, pp. 660-665.

[17] C. T. Kelley, Iterative methods for linear and nonlinear equations, *Frontiers in Applied Mathematics*, Vol. 16, Society for Industrial and Applied Mathematics, Philadelphia, 1995, Chap. 3.

[18] A. A. Amsden and F. H. Harlow, The SMAC Method: A Numerical Technique for Calculating Incompressible Fluid Flows, *Los Alamos Scientific Laboratory of the University of California*, 1970.

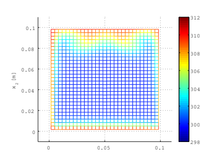

Fig. 2. $t = 2$ [s]

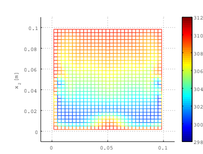

Fig. 3. $t = 6$ [s]

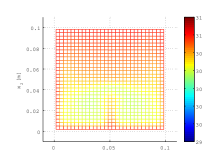

Fig. 4. $t = 10$ [s]

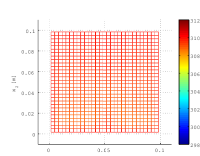

Fig. 5. $t = 20$ [s]

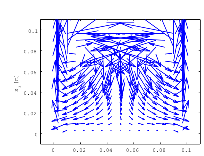

Fig. 6. $t = 2$ [s]

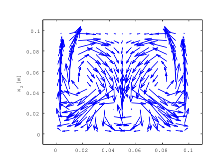

Fig. 7. $t = 10$ [s]

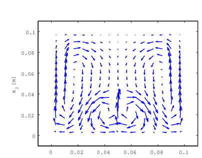

Fig. 8. $t = 20$ [s]

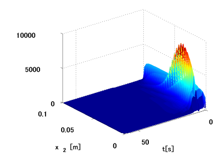

Fig. 9. u_1 $(x_1 = 0.1)$

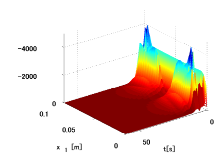

Fig. 10. u_2 $(x_2 = 0)$