Receding Horizon Control With Numerical Solution for Nonlinear Parabolic Partial Differential Equations

Tomoaki Hashimoto, Member, IEEE, Yusuke Yoshioka, and Toshiyuki Ohtsuka, Member, IEEE

Abstract—The optimal control of nonlinear partial differential equations (PDEs) is an open problem with applications that include fluid, thermal, biological, and chemical systems. Receding horizon control is a kind of optimal feedback control, and its performance index has a moving initial time and a moving terminal time. In this study, we develop a design method of receding horizon control for systems described by nonlinear parabolic PDEs. The objective of this study is to develop a novel algorithm for numerically solving the receding horizon control problem for nonlinear parabolic PDEs. The effectiveness of the proposed method is verified by numerical simulations.

Index Terms—Computational algorithm, distributed systems, nonlinear systems, optimal control.

I. INTRODUCTION

Partial differential equations (PDEs) arise from many systems that are characterized by both spatial and temporal variables. The control of nonlinear PDEs is still an open problem as far as general classes of systems are concerned. Control methods for a class of nonlinear PDEs with state-independent coefficients were provided in [1], [2]. However, the control problem for nonlinear PDEs with state-dependent coefficients remains open. In this study, we consider a class of nonlinear PDEs whose coefficients nonlinearly depend on the state variable. To the best of the authors’ knowledge, an optimal controller has not yet been proposed for such a class of nonlinear PDEs. Hence, this study provides a design method of receding horizon control, in which the control performance over a finite future is optimized, for nonlinear parabolic PDEs with state-dependent coefficients. From a practical viewpoint, this study focuses on a numerical method for solving a finite-dimensional control problem instead of considering an infinite-dimensional control problem.

The objective of this study is twofold. In the first part we analytically derive the exact stationary conditions that must be satisfied for a performance index to be optimized. In the second part we provide a novel algorithm for numerically solving the discretized stationary equations in a finite-dimensional setting. Furthermore, the relationship between computational burden and error performance is clarified using the obtained theorem.

The research on receding horizon control (in other words, model predictive control) can be classified into off-line optimization [3] and on-line optimization [4]. This study deals with the on-line optimization problem. Thus, it is not the intent of this study to compare the proposed method with off-line optimization methods. A fast on-line algorithm called CGMRES was proposed in [4] to solve the receding horizon control problem for nonlinear systems described by ordinary differential equations. In this study, we provide a novel framework in which CGMRES is also applicable to solving the receding horizon control problem for nonlinear PDEs.

To achieve real-time optimization, we must solve the stationary conditions as rapidly as possible. Motivated by the fact that the obtained stationary conditions for the optimization problem of nonlinear PDEs have a particular structure with respect to unknown parameters, here we develop an efficient algorithm, which can be used instead of CGMRES, for numerically solving the stationary conditions. More precisely, we propose a simplified method for solving the optimization problem under the assumption that the obtained stationary conditions satisfy a particular structural condition with respect to unknown parameters.

II. RECEDING HORIZON CONTROL

In this section, we consider the receding horizon control problem for a class of systems described by nonlinear parabolic PDEs. We first consider the control problem for systems with boundary control inputs. Next, we analytically derive the stationary conditions that must be satisfied for a performance index to be optimized. Then, we show that the C/GMRES method [4] is applicable to solving the obtained stationary conditions. Moreover, we also consider the control problem of systems with spatially allocated control inputs. We derive the stationary conditions for such a class of systems as well as for systems with boundary control inputs.

A. Boundary Control Inputs

We first consider the following class of nonlinear PDEs:

$$\frac{\partial z(x,t)}{\partial t} = a(z) \frac{\partial}{\partial x} \left( b(z) \frac{\partial z(x,t)}{\partial x} \right) + c(z) \frac{\partial z(x,t)}{\partial x} + d(z) \quad (1)$$
with the Neumann boundary conditions
\[ \frac{\partial z(0, t)}{\partial x} = u_1(t), \quad \frac{\partial z(h, t)}{\partial x} = -u_2(t) \] (2)
and the initial condition \( z(x, 0) = z_0(x) \). Therein, \( a(z), b(z), c(z), \) and \( d(z) : \mathbb{D} \to \mathbb{D} \) are all continuous and differentiable functions with respect to \( z \). For example, a mathematical model of the cooling process of a hot strip mill [5] belongs to such a class of nonlinear PDEs. Let \( E \) be the set defined by \( E := \{ z \in \mathbb{E} \mid z_0(0, t) = u_1(t), z_0(h, t) = -u_2(t) \} \). For notational simplicity, we rewrite system (1) as
\[ \frac{\partial z(x,t)}{\partial t} = \mathcal{F}(z(x,t)) \] (3)
where \( \mathcal{F} : \mathbb{E} \to \mathbb{R} \) is the nonlinear operator defined as
\[ \mathcal{F} := a(z) \frac{\partial}{\partial z} \left( b(z) \frac{\partial z(x,t)}{\partial x} \right) + c(z) \frac{\partial z(x,t)}{\partial x} + d(z) \] (4)
The existence and regularity of solutions, the so-called Cauchy problem, are beyond the scope of this study. Thus, we assume that the solution of (3) is unique and sufficiently smooth. In addition, we assume that \( z(x,t) \) is known at the present time for all \( x \in \Omega \).

In the following, we study the receding horizon control problem for system (3). The control input at each time \( t \) is determined so as to minimize the performance index given by
\begin{align*}
J & = \int_0^{T} L \left[ z(x,t) + p \left( z(x,t) - z_f(x) \right) \right] d\tau \\
& + \int_0^{T} \left( H[z, u_1, u_2, \lambda] + Q[z, \lambda] \right) d\tau \\
& = J \phi \left[ z(x,t + T) \right] + \int_t^{t+T} \left[ H[z, u_1, u_2, \lambda] + Q[z, \lambda] \right] d\tau. 
\end{align*}
(5)

Here, \( H \) denotes the Hamiltonian defined by
\[ H := L + \int_\Omega \lambda(x, \tau) \mathcal{F}(z(x)) dx \]
and \( Q \) is defined by
\[ Q := -\int_\Omega \lambda(x, \tau) \frac{\partial z(x,\tau)}{\partial \tau} dx. \]
(6)

It follows from boundary conditions (2) that
\[ \frac{\partial z(0, t)}{\partial x} = u_1(t), \quad \frac{\partial z(h, t)}{\partial x} = -u_2(t). \] (9)
Substituting (9) into the first term on the right-hand side (RHS) in (8) yields
\[ \int_\Omega \lambda(x, \tau) \frac{\partial z(x,\tau)}{\partial \tau} dx = \left( \sum_{k=1}^2 -\lambda \frac{\partial \mathcal{F}}{\partial z_{x_k}} \delta u_k \right) \]
\[ - \left[ \frac{\partial}{\partial x} \left( \lambda \frac{\partial \mathcal{F}}{\partial z_{x_k}} \right) \delta z \right]_x + \int_\Omega \frac{\partial^2}{\partial x^2} \left( \lambda \frac{\partial \mathcal{F}}{\partial z_{x_k}} \right) \delta z dx. \] (10)

It is also important to note that we have the following integration by parts:
\[ \int_0^{t+T} \left[ \int_\Omega \left( \lambda(x, \tau) \frac{\partial z(x,\tau)}{\partial \tau} dx + \int_\Omega \frac{\partial \lambda}{\partial \tau} \delta z dx \right) \right] d\tau \\
= \int_0^{t+T} \left[ \int_\Omega \left( \lambda(x, \tau) \delta z(x,\tau) dx \right) \right]_t^{t+T} + \int_0^{t+T} \int_\Omega \frac{\partial \lambda}{\partial \tau} \delta z dx d\tau \\
- \int_0^{t+T} \left[ \lambda(x, t + T) \delta z(x, t + T) dx + \int_0^{t+T} \frac{\partial \lambda}{\partial \tau} \delta z dx \right] d\tau. \] (11)

We can set \( \delta z(x, t) = 0 \) in (11) because \( z(x, \tau) \) is fixed at \( \tau = t \). Next, we consider the variation \( \delta \mathcal{F} \) due to the variations \( \delta z, \delta x, \delta z_x, \delta \lambda, \delta u_1, \delta u_2, \) and \( \delta \lambda \). Taking (7), (10), and (11) into account, we obtain the variation in \( J \) as
\begin{align*}
\delta J & = \int_0^{t+T} \left[ \int_\Omega \left( \lambda(x, \tau) \frac{\partial \mathcal{F}(z)}{\partial \tau} dx + \int_\Omega \frac{\partial \lambda}{\partial \tau} \delta z dx \right) \right] d\tau \\
& + \int_0^{t+T} \left[ \int_\Omega \left( \mathcal{F}(z) - \delta \mathcal{F} \right) \delta \lambda dx + \int_\Omega \left( q(z - z_f) \right) \right] d\tau \\
& + \int_0^{t+T} \left[ \left( \frac{\partial \mathcal{F}}{\partial z} + \frac{\partial^2 \mathcal{F}}{\partial z^2} \right) \delta z(x,\tau) dx + \int_\Omega \frac{\partial \lambda}{\partial \tau} \delta z dx \right] d\tau \\
& + \left[ \int_\Omega \left( \frac{\partial \mathcal{F}}{\partial z} \right) \delta z(x,\tau) dx + \int_\Omega \lambda(x, \tau) \frac{\partial \mathcal{F}}{\partial z} \delta z dx \right]_{t}^{t+T} \\
& + \sum_{k=1}^2 \left( \delta u_k(\tau) - \lambda \frac{\partial \mathcal{F}}{\partial z_{x_k}}(\delta u_k) \right) d\tau. 
\end{align*}
(12)

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 have
\[ \frac{\partial z(x,\tau)}{\partial \tau} = \mathcal{F}(z(x,\tau)) \] (13a)
\[ \frac{\partial \lambda(x, \tau)}{\partial \tau} = \lambda(x, \tau + T) \] (13b)
\[ \frac{\partial \lambda(x, \tau)}{\partial \tau} = -\frac{\partial}{\partial x} \left( \lambda \frac{\partial \mathcal{F}}{\partial z} \right) + \int_\Omega \frac{\partial^2}{\partial x^2} \left( \lambda \frac{\partial \mathcal{F}}{\partial z} \right) dx. \] (13c)
and, for \( x = 0, k = 1 \) and \( x = h, k = 2 \), we have
\[ \int_\Omega \frac{\partial^2}{\partial x^2} \left( \lambda \frac{\partial \mathcal{F}}{\partial z} \right) dx = 0. \] (13d)

Conditions (13a)–(13e) are called the stationary conditions, which must be satisfied for the performance index (6) 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.

B. Numerical Solution

To solve stationary conditions (13a)-(13e) using a numerical algorithm, we must discretize (13a)-(13e) into finite difference equations. Here, we divide the space and time into \( M \in \mathbb{N}_+ \) steps and \( N \in \mathbb{N}_+ \) steps, respectively. This means that each step size is given by \( \Delta x := h/(M-1) \) and \( \Delta t := T/(N-1) \). As a result of the discretization, \( z(x, t) \) \( (0 \leq x \leq h; t \leq t + T) \) can be described by \( z_i^j(t) \) \( (i = 1, \ldots, M, j = 1, \ldots, N) \), where the subscripts \( i \) and \( j \) denote space and time, respectively. For other variables, we adopt such notation without explanation. Let \( z_i^j(t) \in \mathbb{R}^M \) and \( \lambda_i^j(t) \in \mathbb{R}^M \) denote \( z_i^j(t) := [z_{1i}^j(t), \ldots, z_{Mi}^j(t)]' \) and \( \lambda_i^j(t) := \lambda_{1i}^j(t), \ldots, \lambda_{Mi}^j(t) \)' respectively. Likewise, let the desired state \( z_f(x) \) be discretized by \( z_f := [z_{f1}, \ldots, z_{fM}]' \in \mathbb{R}^M \). Let \( u_k^j(t) \) \( (t \leq t + T) \) be discretized by \( u_k^j(t) := u_k^j, (j = 1, \ldots, N) \) for \( k = 1, 2, \ldots \), where the subscript \( j \) denotes time over the predictive horizon and \( t \) denotes real time. Let \( u_i^j(t) \in \mathbb{R}^M \) denote space and \( z_i^j(t) \) can be described by (14a). Then, the predictive horizon recedes as real time \( t \) is increased, for \( k = 1, 2, \ldots \), where the subscript \( j \) denotes time over the horizon \( t \leq t + T \) as follows. For \( j = 1, \ldots, N - 1 \), we have

\[
\begin{align*}
z_{j+1}^i(t) &= F (z_j^i(t), u_j^i(t)) \\
p (z_N^i(t) - z_f) &= \lambda_N^i(t) \\
\lambda_j^i(t) &= G (z_{j+1}^i(t), z_j^{i+1}(t))
\end{align*}
\]

where \( F: \mathbb{R}^M \times \mathbb{R}^M \to \mathbb{R}^M \) and \( G: \mathbb{R}^M \times \mathbb{R}^M \to \mathbb{R}^M \) denote nonlinear functions. Note that boundary conditions (2) are also discretized and employed in (14a). Moreover, the equations obtained by discretizing (13c) and (13d) are unified into (14c).

For each \( t \), we obtain the optimal input \( u_j^i(t) \) over the predictive horizon \( j = 1, \ldots, N \) by solving stationary conditions (14a)-(14d). Then, only the first input \( u_1^i(t) \) 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 (14a)-(14d) within the sampling period \( \Delta t \).

Let \( U(t) \in \mathbb{R}^N \) be defined by

\[
U(t) := \left[ u_1^i(t), u_2^i(t), \ldots, u_N^i(t) \right]'.
\]

For the present state \( z_1(t) \) and an initial solution \( U(t); z_1(t)_{i=1, \ldots, N} \) is calculated recursively from \( j = 1 \) to \( N \) by (14a). Then, the terminal costate \( \lambda_N(t) \) is determined by (14b). Consequently, \( \lambda_j^i(t)_{i=1, \ldots, N} \) is also calculated recursively from \( j = N \) to \( j = 1 \) by (14c). Since \( z_j^i(t) \) and \( \lambda_j^i(t) \) are determined by \( z_i^j(t) \) and \( U_j^i(t) \), through (14a)-(14c), (14d) can be regarded as the single equation

\[
H \left( U(t), z_1(t), t \right) := H_1', H_2, \ldots, H_N' \right] \in \mathbb{R}^N
\]

where \( H_{[j]} = 1, \ldots, N \) is defined as

\[
H_{[j]} := \left[ egin{array}{c}
r u_{1j} - \lambda_1 \cdot g(a(z_{1j}), b(z_{1j})) \\
r u_{2j} - \lambda_2 \cdot g(a(z_{2j}), b(z_{2j})) \\
\end{array} \right] \in \mathbb{R}^2.
\]

For given \( z(t) \) and \( U(t) \), \( H \) is not necessarily equal to zero. \( H \) is used to evaluate the optimality performance. If \( |H| = 0 \) is satisfied for given \( z(t) \) and \( U(t) \), then the stationary conditions are satisfied. Several algorithms have been developed such that \( |H| \) can be decreased by suitably updating \( U(t) \). A conventional way of updating \( U(t) \) is to replace \( U(t) \) with \( U(t) + \alpha \nabla H \), which is known as the steepest descent method, where \( \alpha \) is the steepest descent direction and \( \alpha \) is the step length satisfying the Armijo condition [7]. For Newton’s method, \( \alpha \) is given by the Hessian instead of the gradient. However, these methods are computationally expensive, and it was shown that the C/GMRES algorithm [4] is not only faster but also more numerically robust than such conventional algorithms. In the following, a brief description of the C/GMRES method applied to this problem is provided.

Instead of solving \( H(U(t), z_1(t), t) = 0 \) itself at each time by an iterative method such as the steepest descent method or Newton’s method, we find the derivative of \( H(U(t), z_1(t), t) = 0 \) is satisfied identically. Namely, we determine \( U(t) \) such that

\[
H(U(t), z_1(t), t) = -\xi H(U(t), z_1(t), t)
\]

where \( \xi \) is a positive constant introduced to stabilize \( H - 0 \). \( \xi \) is a design parameter to determine the convergence rate of \( |H(U(t)| \).

It was shown in [4] that if we choose \( \xi = 1/2 \Delta t \), then the stability of \( H = 0 \) is guaranteed under several approximations such as the forward difference approximation, truncation in GMRES, and numerical Euler integration of \( U \). Also, we can empirically adjust \( \xi \) to achieve a better performance.

By total differentiation, we have

\[
\frac{\partial H}{\partial U}(U(t),z_1(t),t) = -\xi \frac{\partial H}{\partial z_1}(U(t),z_1(t),t) - \frac{\partial H}{\partial t}(U(t),z_1(t),t)
\]

which can be regarded as a linear algebraic equation with coefficient matrix \( \partial H/\partial U \), which can be used to determine \( U \) for given \( z_1, z_2, \ldots, t \). Then, if the Jacobian \( \partial H/\partial U \) is nonsingular, we obtain the following differential equation for \( U(t) \):

\[
\dot{U}(t) = \left( \frac{\partial H}{\partial U} \right)^{-1} \left( -\xi \frac{\partial H}{\partial z_1}(U(t),z_1(t),t) - \frac{\partial H}{\partial t}(U(t),z_1(t),t) \right)
\]

We can update the solution \( U(t) \) of \( H(U(t), z_1(t), t) = 0 \) without using an iterative optimization method by integrating (19) in real time as, for example, \( U(t + \Delta t) = U(t) + \dot{U}(t) \Delta t \). This approach is a type of continuation method [8] in the sense that the solution curve \( U(t) \) is traced by integrating a differential equation. From the computational viewpoint, the differential (19) still involves expensive operations, that is, the Jacobians \( \partial H/\partial U \), \( \partial H/\partial z_1 \), and \( \partial H/\partial t \) and the linear algebraic equation associated with \( \partial H/\partial U \). To reduce the computational cost of the Jacobians and the linear equation, we employ two techniques: the forward difference approximation for the products of Jacobians and vectors, and the GMRES method [9] for the linear algebraic equation. Using the forward difference approximation, we can obtain a linear equation with respect to \( U \). Then, we can apply the GMRES algorithm [9] to find the solution \( U \) of the linear equation. Consequently, \( U(t) \) can be updated so that \( H = 0 \) is stabilized. More detailed information about the implementation of C/GMRES is provided in [4].

C. Spatially Allocated Control Inputs

In many systems, there are restrictions on the allocation of control actuators. Taking this into account, we introduce space-dependent coefficients of control inputs, \( g_k(x) \); \( \forall k \in K = \{1, \ldots, v\} \), where \( v \in \mathbb{N}_+ \) denotes the number of control inputs. We assume that all \( g_k(x) \) are integrable functions with respect to \( x \). In this subsection, we consider the following class of nonlinear PDEs:

\[
\frac{\partial z(x,t)}{\partial t} = F(z(x,t)) + \sum_{k=1}^{v} g_k(x)u_k(t)
\]
with the Neumann boundary condition
\[
\frac{\partial z(x,t)}{\partial x} = 0 \quad \text{for} \quad x \in \partial \Omega, \quad \forall t \in \mathbb{R}_+.
\]
and the initial condition \( z(x,0) = z_0(x) \), where \( \mathcal{F} \) is defined in (4).

In the following, we study the receding horizon control problem for (20). The control input at each time \( t \) is determined so as to minimize performance index \( J \), given by (5), where \( H \) is replaced with
\[
L = \frac{1}{2} \left( \int_{\Omega} q(z(x,t), z_t(x)) \, dx + \sum_{k=1}^n r_{u_k}(t) \right).
\]

The optimization problem of \( J \) subject to constraint (20) can be reduced to the minimization of performance index \( \tilde{J} \), given by (6), where \( H \) is replaced with
\[
\tilde{H} = L + \int_{\Omega} \lambda(x, \tau) \left( \mathcal{F}(z) + \sum_{k=1}^n g_k(x) u_k(\tau) \right) \, dx.
\]

Note that, from boundary condition (21), we have the following:
\[
\frac{\partial z(x,t)}{\partial x} = 0 \quad \text{for} \quad x \in \partial \Omega, \quad \forall t \in \mathbb{R}_+.
\]

Substituting (24) into (8) yields
\[
\int_0^\lambda \frac{\partial \mathcal{F}}{\partial z_{x_k}} b_{x_{x_k}} \, dx = - \left[ \frac{\partial}{\partial x} \left( \lambda \frac{\partial \mathcal{F}}{\partial z_{x_k}} \right) \right] - \int_0^\lambda \frac{\partial^2 \mathcal{F}}{\partial z_{x_k}} \, dz \, dx.
\]

Applying integration by parts using (7), (11), and (25) to the computation of \( \tilde{J} \) similarly to in Section II-A, we obtain the following stationary conditions: For \( x \in \Omega \) and \( t \leq \tau \leq t + T \), in addition to (13b), (13c), and (20), we have
\[
r_{u_k}(\tau) + \int_0^\lambda \lambda(x, \tau) g_k(x) \, dx = 0, \quad (k = 1, \ldots, n)
\]
and, for \( x \in \partial \Omega \) and \( t \leq \tau \leq t + T \), we have
\[
\left( \lambda \frac{\partial \mathcal{F}}{\partial z_{x_k}} \right) - \frac{\partial}{\partial x} \left( \frac{\partial \mathcal{F}}{\partial z_{x_k}} \right) = 0.
\]

Applying the finite difference method, we obtain the following discretized stationary conditions: For \( j = 1, \ldots, N - 1 \), in addition to (14a)–(14c), we have
\[
r_{u_k}(j) + \sum_{i=1}^M \lambda_{i,j} g_k(x_i) x_j = 0.
\]

Note that the equations obtained by discretizing (13c) and (27) are unified into (14c). The subsequent procedure for solving the obtained optimization conditions is similar to that provided in Section II-B. Hence, it is omitted here.

Next, we consider the following Dirichlet boundary condition instead of the Neumann boundary condition (21):
\[
z(x, t) = z_\sigma \quad \text{for} \quad x \in \partial \Omega, \quad \forall t \in \mathbb{R}_+.
\]

where \( z_\sigma \in \mathbb{R} \) is a constant. In this case, considering that \( \partial z(x, t') = 0 \) for \( x \in \partial \Omega, \forall t \in \mathbb{R}_+ \), we similarly obtain the following stationary conditions. For \( x \in \Omega \) and \( t \leq \tau \leq t + T \), we have (13b), (13c), (20), and (26), and for \( x \in \partial \Omega \) and \( t \leq \tau \leq t + T \), we have
\[
\left( \lambda \frac{\partial \mathcal{F}}{\partial z_{x_k}} \right) = 0.
\]

The subsequent procedure for solving the stationary conditions is similar to the above procedure. Hence, it is omitted here.

### III. CONTRACTION MAPPING METHOD

As can be seen in Section II-B, \( z_j(t) \) and \( \lambda_j(t) \) over the horizon \( j = 1, \ldots, N \) are determined from the present state \( z_1(t) \) and a given \( U(t) \) through (14a)–(14c). Hence, the optimization conditions can be regarded as \( \mathbf{H}(U(1), z_1(t), t) = 0 \). In this section, we develop an efficient algorithm, instead of using the C/GMRES algorithm, for solving \( \mathbf{H}(U(1), z_1(t), t) = 0 \). More precisely, here we propose a novel algorithm for solving \( \mathbf{H}(U(1), z_1(t), t) = 0 \) under the assumption that \( \mathbf{H}(U(1), z_1(t), t) \) satisfies particular structural conditions with respect to \( U \). In the following, we assume that \( \mathbf{H} \) is given by:
\[
\mathbf{H}(U(t), z_1(t), t) = \mathbf{A}U(t) - \mathbf{b}(U(t), z_1(t), t)
\]
where \( \mathbf{A} \in \mathbb{R}^{n \times n} \) is a nonsingular constant matrix and \( \mathbf{b} \in \mathbb{R}^n \) is a vector-valued function satisfying the Lipschitz continuous condition, i.e., there exist \( v_1, v_2, v_3 \in \mathbb{R}_+ \) such that \( \mathbf{b} \) satisfies
\[
\| \mathbf{b}(U(t), z_1(t), t) - \mathbf{b}(U(t), z_1(t), t) \| \leq v_1 \| U(t) - \tilde{U}(t) \| + v_2 \| z_1(t) - \tilde{z}_1(t) \| + v_3 \| t - t \|.
\]

In addition, we assume that there exists \( v_4 \in \mathbb{R}_+ \) such that
\[
\| z_1(t) - \tilde{z}_1(t) \| \leq v_4 \| t - t \|.
\]

Motivated by the fact that the optimal control problem for nonlinear parabolic PDEs can be reduced to solving the stationary condition represented by (31), we focus on a numerical algorithm for solving stationary condition (31) that has a structural restriction. Compared with the C/GMRES method [4], the proposed algorithm has the disadvantage of limited applicability but the advantage of a smaller computational burden.

Let \( \mathbf{P} \in \mathbb{R}^n \) be defined by
\[
\mathbf{P}(U, z_1, t) = \mathbf{A}^{-1} \mathbf{b}(U, z_1, t).
\]

Here we adopt the following notations:
\[
\mathbf{P} \circ \mathbf{P}(U, z_1, t) = \mathbf{P}(\mathbf{P}(U, z_1, t), z_1, t)
\]
\[
\mathbf{P}^k(U, z_1, t) = \mathbf{P} \circ \cdots \circ \mathbf{P}(U, z_1, t).
\]

Let \( \mathbf{H}(U(t), z_1(t), t) \) and \( \mathbf{H}(U(t - \Delta t), z_1(t - \Delta t), t - \Delta t) \) be denoted by \( \mathbf{H}_{(t)} \) and \( \mathbf{H}_{(t - \Delta t)} \), respectively, for the sake of simplicity. Suppose that \( \mathbf{U} \) is updated as
\[
\mathbf{U}(t) = \mathbf{P}(\mathbf{U}(t - \Delta t), z_1(t), t)
\]
where \( t = \Delta t, 2\Delta t, \ldots, j\Delta t \) for \( j \in \mathbb{N}_+ \), and \( k \) is a design parameter. Then, we can state the following theorem.

**Theorem 1:** Assuming that
\[
v_4 \| \mathbf{A}^{-1} \| < 1
\]
is satisfied, then \( \| \mathbf{H}_{(t)} \| \) is ultimately bounded as follows:
\[
\| \mathbf{H}_{(t)} \| \leq \varepsilon \quad \Rightarrow \quad \| \mathbf{H}_{(t)} \| \leq \varepsilon
\]
\[
\| \mathbf{H}_{(t - \Delta t)} \| > \varepsilon \quad \Rightarrow \quad \| \mathbf{H}_{(t)} \| < \| \mathbf{H}_{(t - \Delta t)} \|
\]
where $\varepsilon$ is given by
\[
\varepsilon = \frac{(v_1 \| A^{-1} \|)^2}{1 - (v_1 \| A^{-1} \|)^2} (l_2 u_4 + \varepsilon_3) \Delta t.
\] (39)

Note that $\| H_{i,t} \|$ is monotonically decreasing whenever $\| H_{i,t} \| > \varepsilon$.

**Proof:** Substituting (35) into (31), we have
\[
\begin{align*}
H(U(t), \mathbf{z}_1, t) &= \mathbf{A} P^k(U(t, \mathbf{z}_1, t)) - b \circ P^k(U(t, \mathbf{z}_1, t)) \\
&= \mathbf{A} P^k(U(t, \mathbf{z}_1, t)) - b \circ P^k(U(t, \mathbf{z}_1, t)).
\end{align*}
\] (40)

For the sake of simplicity, let $\mathbf{U}(t + \Delta t)$ and $\mathbf{z}_1(t + \Delta t)$ be denoted by $\mathbf{U}_{(t+\Delta t)}$ and $\mathbf{z}_{1(t+\Delta t)}$, respectively.

For fixed $\mathbf{z}_1$ and $t$, it follows from (34) that
\[
\begin{align*}
H(U(t), \mathbf{z}_1, t) &= \mathbf{A} P^k(U_{(t+\Delta t)}, \mathbf{z}_1, t) - b \circ P^k(U_{(t+\Delta t)}, \mathbf{z}_1, t) \\
&= \mathbf{A} P^k(U_{(t+\Delta t)}, \mathbf{z}_1, t) - b \circ P^k(U_{(t+\Delta t)}, \mathbf{z}_1, t).
\end{align*}
\] (41)

Applying (32) to (41) repeatedly for fixed $\mathbf{z}_1$ and $t$, we have
\[
\begin{align*}
\| H(U(t), \mathbf{z}_1, t) \| &= \| \mathbf{A} P^k(U_{(t+\Delta t)}, \mathbf{z}_1, t) - b \circ P^k(U_{(t+\Delta t)}, \mathbf{z}_1, t) \| \\
&\leq v_1 \| \mathbf{A}^k - v_1 \| \| U_{(t+\Delta t)} - P(U_{(t+\Delta t)}, \mathbf{z}_1, t) \| \\
&\leq v_1 \| \mathbf{A}^{-1} \| \| U_{(t+\Delta t)} - P(U_{(t+\Delta t)}, \mathbf{z}_1, t) \| \\
&\leq v_1 \| \mathbf{A}^{-1} \| \| U_{(t+\Delta t)} - P(U_{(t+\Delta t)}, \mathbf{z}_1, t) \|.
\end{align*}
\] (42)

Now, observing that
\[
\begin{align*}
U_{(t+\Delta t)} - P(U_{(t+\Delta t)}, \mathbf{z}_1, t) &= \mathbf{A}^{-1} H(U_{(t+\Delta t)}) \\
&+ P(U_{(t+\Delta t)}, \mathbf{z}_1, t) - P(U_{(t+\Delta t)}, \mathbf{z}_1, t)
\end{align*}
\] (43)

it follows that
\[
\begin{align*}
\| U_{(t+\Delta t)} - P(U_{(t+\Delta t)}, \mathbf{z}_1, t) \| &\leq \| \mathbf{A}^{-1} \| \| H(U_{(t+\Delta t)}) \| + \| \mathbf{A}^{-1} \| \| v_2 \| (l_2 u_4 + \varepsilon_3) \Delta t \\
&\leq \| \mathbf{A}^{-1} \| \| H(U_{(t+\Delta t)}) \| + \| v_2 \| (l_2 u_4 + \varepsilon_3) \Delta t.
\end{align*}
\] (44)

Substituting (44) into (42), we have
\[
\| H(U_{(t+\Delta t)}) \| \leq \| v_1 \| \| \mathbf{A}^{-1} \| \| H(U_{(t+\Delta t)}) \| + \| v_2 \| (l_2 u_4 + \varepsilon_3) \Delta t.
\] (45)

Let $\alpha_1$ and $\alpha_2$ be defined by
\[
\alpha_1 := \| v_1 \| \| \mathbf{A}^{-1} \| \| H(U_{(t+\Delta t)}) \|, \quad \alpha_2 := \| v_2 \| (l_2 u_4 + \varepsilon_3) \Delta t.
\]

Suppose that $\| H(U_{(t+\Delta t)}) \|$ converges as $\| H(U_{(t+\Delta t)}) \| < \varepsilon$ and $\varepsilon < \alpha_1 + \alpha_2$ hold. Then, $\varepsilon$ has a nonnegative solution (39) under the assumption that $\| v_1 \| \| \mathbf{A}^{-1} \| < 1$. Moreover, it follows from (45) that:
\[
\begin{align*}
\| H(U_{(t+\Delta t)}) \| &\leq \varepsilon \Rightarrow \| H(U_{(t+\Delta t)}) \| < \alpha_1 + \alpha_2 = \varepsilon \\
\| H(U_{(t+\Delta t)}) \| &> \varepsilon \Rightarrow \| H(U_{(t+\Delta t)}) \| < \alpha_1 + (1 - \alpha_1) \varepsilon < \alpha_1 + \alpha_2 \\
&< \| H(U_{(t+\Delta t)}) \| + (1 - \alpha_1) \| H(U_{(t+\Delta t)}) \|
\end{align*}
\]

This completes the proof.

---

Note that Theorem 1 enables the error analysis of the proposed algorithm (35). It is worth noting that the upper bound $\varepsilon$ of $\| H(U_{(t+\Delta t)}) \|$ converges to zero as $k$ increases to infinity. Provided that $v_1 \| A^{-1} \| < 1$ is satisfied. From this viewpoint, the proposed method provides a satisfactory tradeoff between computational burden and error performance through the selection of design parameter $k$.

Consider that
\[
\| P(U, \mathbf{z}_1, t) - P(\hat{U}, \mathbf{z}_1, t) \| \leq \{ v_1 \| A^{-1} \| \} \| U - \hat{U} \|
\] (46)

we see from $v_1 \| A^{-1} \| < 1$ that $P$ is a contraction map with respect to $U$. Hence, we call the proposed algorithm the contraction mapping method. When the obtained stationary conditions satisfy (31), (32), (33), and (36), we can find the optimal control input by using the contraction mapping method instead of the C/GMRES method.

---

**IV. NUMERICAL SIMULATIONS**

In this section, we provide an illustrative example to verify the effectiveness of the proposed method. We consider
\[
\frac{\partial x}{\partial t} = \frac{\partial}{\partial x} \left( \frac{0.2}{x} \frac{\partial x}{\partial x} \right) + \frac{2}{1} g(x) u_k(t)
\] (47)

with boundary conditions (21) and the initial condition $x_0(x) = 1.5 \sin(2\pi x) + 2$. Fig. 1 shows an example of $g(x)$ that the control input $u_k$ is only employed at point $x_1$ through a discretizing approximation. Thus, $g_1$ and $g_2$ are given here such that $u_1$ and $u_2$ can be employed at $x = (\Delta x/2)/3$ and $x = (\Delta x/2)/3$, respectively. The desired state $x_f(x)$ is set as $x_f(x) = 1$ for all $x \in \Omega$. Owing to the initialization of the optimal solution $U(0)$, the length of the horizon is chosen so that $T(0) = 0$ and $T(0) = 0.05$ as $t \to \infty$, namely, $T(t) = 0.05(1 - e^{-0.05t})$. Other parameters employed in the numerical simulations are as follows: $h = 1$, $\Delta x = 0.005$, $\Delta t = 0.005$, $M = 200$, $N = 10$, $k = 1$, and $[p, q, r] = [10, 5000, 1]$. Figs. 2 and 3 show the time history of the state variable $x(x, t)$ of (47) without control inputs and with control inputs, respectively. Figs. 4 and 5 show the time history of the control inputs and the optimality error, respectively. The figures reveal the effectiveness of the proposed method.

The simulation was performed on a laptop computer (Panasonic CF-S9, CPU: Intel(R) Core(TM) i5, 2.4 GHz—Memory: 3.4 GB—OS: Windows 7, 32bit—Software: Matlab). The average computational time per update (one control cycle) and the average optimality error for the proposed method and the C/GMRES method are listed in Table I. The iteration number of GMRES and a design parameter $k$ are set as 1 and 3, respectively. It can be seen that the optimality error of the proposed method with $k = 1$ is larger than that of the C/GMRES method, but the optimality error of the proposed method with $k = 2$ is smaller than that of the C/GMRES method.

Moreover, the computational time of the proposed method is smaller.
TABLE I

<table>
<thead>
<tr>
<th></th>
<th>Average Computational Time [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/GMRES method [4]</td>
<td>49.4</td>
</tr>
<tr>
<td>Proposed method ($k = 1$)</td>
<td>8.8</td>
</tr>
<tr>
<td>Proposed method ($k = 2$)</td>
<td>18.4</td>
</tr>
<tr>
<td>Average Optimality Error</td>
<td></td>
</tr>
<tr>
<td>C/GMRES method [4]</td>
<td>$1.70 \times 10^{-4}$</td>
</tr>
<tr>
<td>Proposed method ($k = 1$)</td>
<td>$5.14 \times 10^{-4}$</td>
</tr>
<tr>
<td>Proposed method ($k = 2$)</td>
<td>$0.53 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

than that of the C/GMRES method. Consequently, the effectiveness of the proposed method was verified by these simulations.

V. CONCLUSION

In this technical note, we investigated the optimal control problem for a class of nonlinear parabolic PDEs with state-dependent coefficients. We first formulated a novel framework for designing receding horizon control for this class of systems. On the basis of the variational principle, we derived the stationary conditions that must be satisfied for the control performance over a finite future to be optimized. Next, the obtained stationary conditions were reduced to finite difference equations to enable them to be numerically solved, because they usually cannot be solved analytically. It was shown that the C/GMRES method [4] is applicable to solving the receding horizon control problem for nonlinear parabolic PDEs upon suitable formulation and modification. Motivated by the fact that the obtained stationary conditions for the optimal control problem of nonlinear parabolic PDEs satisfy particular structural conditions with respect to unknown parameters, we proposed a simplified algorithm with a lower computational burden than the C/GMRES algorithm. Compared with the C/GMRES method, the proposed algorithm has the disadvantage of limited applicability but the advantage of a smaller computational burden. Furthermore, it was shown that the proposed algorithm provides a satisfactory tradeoff between computational burden and error performance. Finally, the effectiveness of the proposed method was verified by numerical simulation.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the associate editor for detailed comments on the technical note which helped to improve the clarity of the manuscript.

REFERENCES