Received: 13 September 2018 | Revised: 18 March 2019 | Accepted: 6 October 2019

DOI: 10.1002/rnc.4804

RESEARCH ARTICLE

WILEY

A moving switching sequence approach for nonlinear model predictive control of switched systems with state-dependent switches and state jumps

Sotaro Katayama[1] | **Masahiro Doi**[2] | **Toshiyuki Ohtsuka**[1]

[1]Department of System Science, Graduate School of Informatics, Kyoto University, Kyoto, Japan

[2]The Advanced Robot Group, Planning and Management Department, Z-Frontier Division, Frontier Research Center, Toyota Motor Corporation, Tokyo, Japan

Correspondence
Toshiyuki Ohtsuka, Department of System Science, Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan.
Email: ohtsuka@i.kyoto-u.ac.jp

Funding information
JSPS Kakenhi, Grant/Award Number: JP15H02257

Summary

In this paper, we propose an approach for real-time implementation of nonlinear model predictive control (NMPC) for switched systems with state-dependent switches called the *moving switching sequence approach*. In this approach, the switching sequence on the horizon moves to the present time at each time as well as the optimal state trajectory and the optimal control input on the horizon. We assume that the switching sequence is basically invariant until the first predicted switching time reaches the current time or a new switch enters the horizon. This assumption is reasonable in NMPC for systems with state-dependent switches and reduces computational cost significantly compared with the direct optimization of the switching sequence all over the horizon. We update the switching sequence by checking whether an additional switch occurs or not at the last interval of the present switching sequence and whether the actual switch occurs or not between the current time and the next sampling time. We propose an algorithm consisting of two parts: (1) the local optimization of the control input and switching instants by solving the two-point boundary-value problem for the whole horizon under a given switching sequence and (2) the detection of an additional switch and the reconstruction of the solution taking into account the additional switch. We demonstrate the effectiveness of the proposed method through numerical simulations of a compass-like biped walking robot, which contains state-dependent switches and state jumps.

KEYWORDS

biped walking, hybrid systems, nonlinear model predictive control, switched systems

1 | INTRODUCTION

Switched systems are a class of hybrid systems composed of several subsystems and switching laws. Many systems in the real world can be modeled as switched systems and can be roughly divided into two types: systems that switch when the state satisfies some conditions, and systems that switch when the system receives external signals.[1] The former are called state-dependent switches or internally forced switches, and the latter are called state-independent switches or exter-

nally forced switches. For example, we can see state-dependent switches in contacts of mechanical systems with objects or environments[2-5] and logical switching controllers[6-8] and state-independent switches in gear shifts of automobiles,[9] electrical circuit systems,[10,11] and process systems.[12,13] In this paper, we focus on control of switched systems with state-dependent switches.

It is difficult to apply conventional control methods directly to switched systems with state-dependent switches, particularly when they contain nonlinear subsystems. Nonlinear model predictive control (NMPC)[14-16] is the only generic method that can control such systems in practice. In NMPC, an open-loop optimal control problem over a finite future is solved at each time on the basis of the current state and the system's model, and the actual control input to the system is given by the initial value of the optimal control input. Even if the system contains discrete events such as switches or state jumps, we can realize NMPC in consideration of future discrete events as long as we can solve the finite horizon nonlinear optimal control problem (NOCP) within a given sampling period. However, it is difficult to solve the finite horizon NOCP for the switched systems with state-dependent switches in a short computational time because the control input and sequence of the active subsystems complicatedly depend on each other.[17]

In previous studies, optimal control of systems with state-independent switches and that of systems with state-dependent switches are often considered in the same framework, and many different optimization approaches have been studied. Mixed-integer programming (MIP) is a classical method for optimization problem with a kind of switches and numerical methods based on MIP are successfully applied to systems with long sampling period, eg, process systems.[18,19] Two-stage methods[20-22] define two independent optimal control problems (one for the sequence of active subsystems over the horizon, and the other for control input and switching instants) and solve them alternately by conventional numerical methods for optimization. The algebraic geometry method[23,24] applies cylindrical algebraic decomposition to the NOCP for polynomial switched systems. The embedded method[25,26] converts the optimal control problem for switched systems into a conventional continuous optimal control problem for a larger set of systems by introducing variables corresponding to mode activation. Although these methods successfully optimize the sequence of active subsystems and control input simultaneously, it is difficult to realize NMPC by using these methods when the switched system contains nonlinear subsystems whose state equations are complex, and the sampling period is short because these methods still take long computation time. These studies require seeking the switching sequence, sequence of active subsystems, all over the horizon of NOCP, which increases the computational time significantly compared with the optimal control problem for systems without switches. On the other hand, the minimum-principle for the switched or hybrid systems has been studied for a long time,[27,28] and efficient numerical methods combining it with conventional optimization algorithms, such as dynamic programming or gradient-descent methods under a fixed sequence of the active subsystems, have been developed.[17,29-35] However, these methods increase the computational cost a lot compared with the optimal control problem or NMPC for a system without any switches. These methods have to find an optimal (or at least a feasible) sequence of active subsystems on the horizon. For this purpose, they have to solve the optimal control problems for all possible sequence of active subsystems, which obviously takes much more computational time than solving the optimal control problem for only one sequence of active subsystems.

In this paper, we propose an approach for real-time implementation of NMPC for switched systems with state-dependent switches called the *moving switching sequence approach*, in which the switching sequence on the horizon moves to the present time at each time as well as the optimal state trajectory and the optimal control input on the horizon. We assume that the switching sequence is basically invariant until the first predicted switching time reaches the current time or a new switch enters the horizon. This assumption is reasonable in NMPC for systems with state-dependent switches and reduces computational cost significantly compared with directly optimizing the switching sequence all over the horizon. In the proposed method, we update the switching sequence by checking whether an additional switch occurs or not at the last interval of the present switching sequence and whether the actual switch will occur or not between the current sampling time and the next sampling time. We check the former by computing the state trajectory on the horizon on the basis of the current solution and the current switching sequence. If we detect an additional switch on the state trajectory, we reconstruct the solution of NMPC in consideration of the additional switch. We check the latter by evaluating the switching instants after the solution is updated. If the instant of the first switch in the switching sequence after the solution is updated becomes less than the next sampling time, we predict that the actual switch occurs between the current sampling time and the next sampling time and then remove variables related to the switch from the solution. We propose an algorithm of the moving switching sequence approach consisting of two parts: (1) the local optimization of the control input and switching instants by solving the two-point boundary-value problem (TPBVP) for the whole horizon under a given switching sequence and (2) the detection of an additional switch and the reconstruction of the solution by solving a reduced optimal control problem defined only for additional variables with respect to an additional switch.

The former takes almost the same computational time as the NOCP for systems without switches, and the latter can be achieved at sufficiently small computational burden. Therefore, the computational time of the proposed method increases only slightly from that of NMPC for a system without switches.

In solving the TPBVP, we can apply conventional numerical methods for implementing NMPC in real time.[36-42] We adopt the continuation/generalized minimum residual (C/GMRES) method,[37] which achieves fast computation by tracking the solution at each time without approximation for state equations and cost functions as long as the sampling period is set sufficiently short. By utilizing the C/GMRES method for the TPBVP containing switches, we update the control input and switching instants simultaneously at each sampling time in a short computational time. In reconstructing the solution for an additional switch, we obtain variables with respect to the switch by solving the reduced optimal control problem only for variables with respect to the additional active subsystem using conventional numerical methods, eg, Newton's method. Note that there are studies using the C/GMRES method that implicitly make the same assumption about the invariance of switching sequence for NMPC for a class of switched systems with state-dependent switches.[43-45] For example, Yamakita et al extended the C/GMRES method to mechanical systems with collisions by handling the optimality conditions approximately.[43,44] However, that approximation generates errors from the optimal control, and an undesirable oscillation of the control input is observed in their simulations. They introduce an integrator to avoid that oscillation, which does not guarantee the optimality of the solution. To treat the optimality conditions more appropriately with the C/GMRES method, we proposed a penalty function method.[45] However, this method can make numerical computation unstable due to the penalty function. These approximations are introduced to avoid directly treating changes in the number of optimization variables caused by the addition of a new switch. In contrast, the moving switching sequence approach does not introduce such approximations because this method obtains the additional variables efficiently when the number of switches on the horizon changes.

The rest of this paper is composed as follows. In Section 2, we describe a class of switched systems with state-dependent switches. In Section 3, we derive the optimality conditions and define the TPBVP for that system. In Section 4, we describe the C/GMRES method for TPBVP formulated in Section 3, detection of an additional switch and reconstruction of the solution for the additional switch, and the algorithm of the moving switching sequence approach. In Section 5, we present a numerical simulation of a compass-like robot walking for two cases, ie, a nominal case without any disturbances and a case with impulse disturbances, using the proposed method and demonstrate the method's effectiveness. In Section 6, we conclude our paper.

2 | SWITCHED SYSTEMS WITH STATE-DEPENDENT SWITCHES

We consider a switched system consisting of subsystems

$$\begin{aligned} &\dot{x}(t) = f_{q(t)}(x(t), u(t)), \quad f_q : \chi_q \times \mathbb{R}^m \to \mathbb{R}^n, \\ &q(t) \in Q := \{1, 2, \ldots, M\}, \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ denotes the state vector, $u(t) \in \mathbb{R}^m$ denotes the control input vector, $\chi_q \subseteq \mathbb{R}^n$ denotes the domain of $f_q(x, u)$ for x, and M denotes a positive integer. We represent that *subsystem $q(t)$ is active* or *active subsystem is $q(t)$* if the state is governed by $\dot{x}(t) = f_{q(t)}(x(t), u(t))$. We also consider the switching sets $\Psi_{q_1,q_2} \subseteq \chi_{q_1}$ that represent the condition of state-dependent switches

$$\Psi_{q_1,q_2} = \left\{ x \,|\, \psi_{q_1,q_2}(x) = 0, \psi_{q_1,q_2} : \chi_{q_1} \to \mathbb{R}^{l_{q_1,q_2}} \right\}, \quad q_1, q_2 \in M. \tag{2}$$

This means that, if subsystem q_1 is active and $x \in \chi_{q_1}$ reaches Ψ_{q_1,q_2}, that is, if $x \in \chi_{q_1}$ satisfies $\psi_{q_1,q_2}(x) = 0$, then the active subsystem switches to q_2. We also consider the state changes discontinuously, ie, a *state jump* occurs at the same time as the switch, as

$$x^+ = \gamma_{q_1,q_2}(x^-), \quad \gamma_{q_1,q_2} : \mathbb{R}^n \to \mathbb{R}^n, \quad q_1, q_2 \in Q, \tag{3}$$

where $x^- \in \Psi_{q_1,q_2}$ denotes the state just before the state jump, and $x^+ \in \mathbb{R}^n$ denotes the state just after the state jump. Note that, if x satisfies $\psi_{q,q}(x) = 0$ for some $q \in Q$, just the state jump $x^+ = \gamma_{q,q}(x^-)$ occurs without a switch of the active subsystems. We make the following assumptions about $f_q(x, u)$, $\psi_{q_1,q_2}(x)$, and $\gamma_{q_1,q_2}(x)$.

Assumption 1. $f_q(x,u)$ are continuously differentiable for all $x \in \chi_q$, for all $u \in \mathbb{R}^m$, and for all $q \in Q$, and $\psi_{q_1,q_2}(x)$ and $\gamma_{q_1,q_2}(x)$ are also continuously differentiable for all $x \in \mathbb{R}^n$ and for all $q_1, q_2 \in Q$.

Assumption 2. $\gamma_{q_1,q_2}(x)$ is uniquely and explicitly determined for all $x \in \chi_{q_1}$ and for all $q_1, q_2 \in Q$.

Assumption 3. $\gamma_{q_1,q_2}(x) \in \chi_{q_2}$ for all $x \in \chi_{q_1}$ and for all $q_1, q_2 \in Q$.

Assumption 4. $\gamma_{q_1,q_2}(x) \notin \Psi_{q_2,q_i}$ for all $x \in \Psi_{q_1,q_2}$ and for all $q_1, q_2, q_i \in Q$.

Note that Assumption 4 implies that the length of a time interval in which any subsystem is active is not zero, ie, state jumps do not occur in a row. That is, situations that are too complicated such as infinite loops of state jumps are not considered.

3 | TWO-POINT BOUNDARY-VALUE PROBLEMS

In NMPC, the optimality conditions, which are the necessary conditions of the optimal control, are formulated into a TPBVP and solved numerically to obtain the optimal control input at each time. The optimal control problem for switched systems with state-dependent switches and state jumps is defined as follows: find $u(t')$ $(t \le t' \le t+T)$ minimizing the cost function

$$J = \varphi_{q(t+T)}(x(t+T)) + \int_t^{t+T} L_{q(t')}\left(x(t'), u(t')\right) dt', \tag{4}$$

subject to (1)-(3), where $\varphi_q(x) \in \mathbb{R}$ denotes the terminal cost for subsystem q and $L_q(x) \in \mathbb{R}$ denotes the stage cost for subsystem q. To derive the optimality conditions, we introduce a sequence of subsystems for $t \in [t, t+T]$, *a switching sequence* defined as

$$\sigma = (q_1, q_2, \ldots, q_m), \tag{5}$$

where $q_k \in Q$ for all $k = 1, \ldots, m$, and introduce switching instants $t_k \in \mathbb{R}$ for $k = 1, \ldots, m-1$ representing the instants of the switch from subsystem q_k to subsystem q_{k+1}, where $0 \le m < \infty$, $t \le t_1 < \cdots < t_{m-1} \le t+T$, and $q_1, \ldots, q_m \in Q$. Note that $t_i \ne t_{i+1}$ because of Assumption 4. This denotes that subsystem q_1 is active at $[t, t_1)$, subsystem q_k is active at (t_{k-1}, t_k) for $k = 2, \ldots, m-1$, and subsystem q_m is active at $(t_{m-1}, t+T]$. Figure 1 illustrates the switching sequence and a state trajectory on the horizon. We make the following assumption about the solution of the NOCP for systems with state-dependent switches.

Assumption 5. The optimal control input u and switching sequence σ of a finite length exist.

Suppose that the switching sequence on the horizon $t' \in [t, t+T]$ is given as $\sigma = (q_1, q_2, \ldots, q_m)$, where $q_1, \ldots, q_m \in Q$ and the instants of the switch from q_k to q_{k+1} are given as t_k for all $k = 1, \ldots, m-1$. The state trajectory is then given as

$$\frac{d}{dt'}x(t') = f_{q_1}\left(x(t'), u(t')\right) \quad (t \le t' < t_1), \tag{6}$$

$$\psi_{q_1,q_2}\left(x\left(t_1^-\right)\right) = 0, \tag{7}$$

$$x\left(t_1^+\right) = \gamma_{q_1,q_2}\left(x\left(t_1^-\right)\right), \tag{8}$$

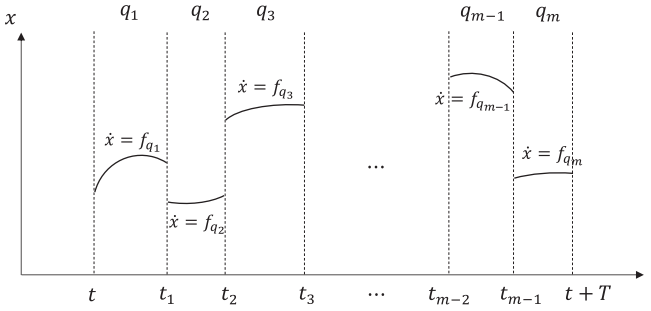

FIGURE 1 The switching sequence $\sigma = (q_1, q_2, \ldots, q_m)$ and a state trajectory on the horizon

$$\left.\begin{aligned} &\frac{d}{dt'}x(t') = f_{q_k}\left(x(t'), u(t')\right) \quad (t_{k-1} < t' < t_k) \\ &\psi_{q_k,q_{k+1}}\left(x\left(t_k^-\right)\right) = 0 \\ &x(t_k^+) = \gamma_{q_k,q_{k+1}}\left(x\left(t_k^-\right)\right) \end{aligned}\right\} \quad \text{for} \quad k = 2, \ldots, m-1, \tag{9}$$

$$\frac{d}{dt'}x(t') = f_{q_m}\left(x(t'), u(t')\right) \quad (t_{m-1} < t' < t+T). \tag{10}$$

The optimal control problem is then defined as finding the optimal control input $u(t')$ $(t \le t' \le t+T)$ minimizing the cost function

$$J = \varphi_{q_m}\left(x(t+T)\right) + \int_{t_{m-1}}^{t_f} L_{q_m}\left(x(t'), u(t')\right) dt' + \sum_{k=2}^{m-1} \int_{t_{k-1}}^{t_k} L_{q_k}\left(x(t'), u(t')\right) dt' + \int_{t}^{t_1} L_{q_1}\left(x(t'), u(t')\right) dt', \tag{11}$$

subject to (6)-(10).

For numerical computation, we discretize the optimal control problem. We divide the horizon $t' \in [t, t+T]$ into N steps, define the time step $\Delta\tau := T/N$, and introduce i_k as an integer satisfying $i_k\Delta\tau \le t_k - t < (i_k+1)\Delta\tau$ for $k = 1, \ldots, m-1$. We also introduce $x_i^*(t)$ for $i = 0, \ldots, N$ and $u_i^*(t)$ for $i = 0, \ldots, N-1$ corresponding to $x(i\Delta\tau + t)$ and $u(i\Delta\tau + t)$. Figure 2 shows an example of the discretization over the horizon, in which $i_k\Delta\tau + t < t_k$ for $k = 1, \ldots, m-2$ and $i_{m-1}\Delta\tau + t = t_{m-1}$. The optimal control problem is then given as follows: find the optimal control input sequence $u_0^*(t), \ldots, u_{N-1}^*(t), u^*(t_1^+), \ldots, u^*(t_{m-1}^+)$ minimizing the discretized cost function

$$\begin{aligned} J = {} & \varphi_{q_m}\left(x_N^*(t)\right) + \sum_{i=i_m+1}^{N-1} L_{q_m}\left(x_i^*(t), u_i^*(t)\right)\Delta\tau + L_{q_m}\left(x^*\left(t_{m-1}^+\right), u^*\left(t_{m-1}^+\right)\right)\left((i_{m-1}+1)\Delta\tau + t - t_{m-1}\right) \\ & + \sum_{k=2}^{m-1}\left\{ L_{q_k}\left(x_{i_k}^*(t), u_{i_k}^*(t)\right)\left(t_k - t - i_k\Delta\tau\right) + \sum_{i=i_{k-1}+1}^{i_k-1} L_{q_k}\left(x_i^*(t), u_i^*(t)\right)\Delta\tau \right. \\ & \left. + L_{q_k}\left(x^*\left(t_{k-1}^+\right), u^*\left(t_{k-1}^+\right)\right)\left((i_{k-1}+1)\Delta\tau + t - t_{k-1}\right)\right\} \\ & + L_{q_1}\left(x_{i_1}^*(t), u_{i_1}^*(t)\right)\left(t_1 - t - i_1\Delta\tau\right) + \sum_{i=0}^{i_1-1} L_{q_1}\left(x_i^*(t), u_i^*(t)\right)\Delta\tau, \end{aligned} \tag{12}$$

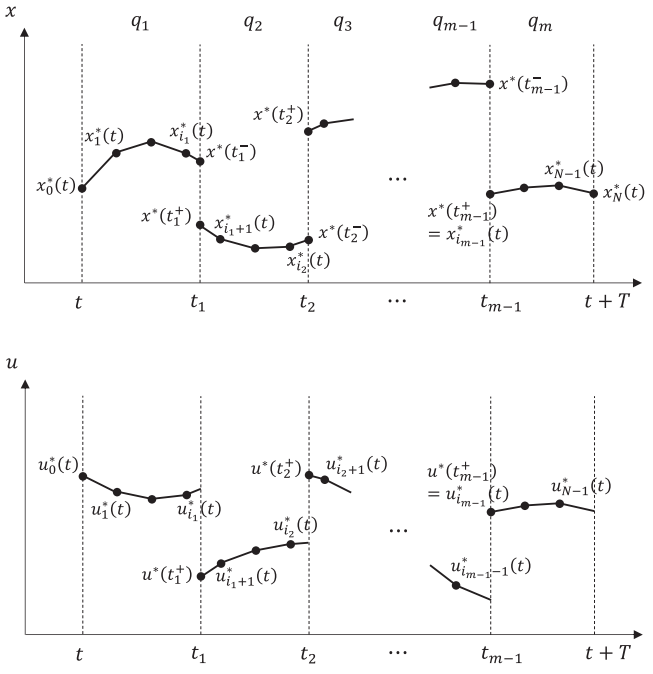

FIGURE 2 An example of the discretization of x and u over the horizon

subject to

$$x_0^*(t) = x(t), \tag{13}$$

$$x_{i+1}^*(t) = x_i^*(t) + f_{q_1}\left(x_i^*(t), u_i^*(t)\right)\Delta\tau, \quad i = 0, \ldots, i_1 - 1, \tag{14}$$

$$x^*(t_1^-) = x_{i_1}^*(t) + f_{q_k}\left(x_{i_1}^*(t), u_{i_1}^*(t)\right)(t_1 - t - i_1\Delta\tau), \tag{15}$$

$$\psi_{q_1,q_2}\left(x^*\left(t_1^-\right)\right) = 0, \tag{16}$$

$$x^*\left(t_1^+\right) = \gamma_{q_1,q_2}\left(x^*\left(t_1^-\right)\right), \tag{17}$$

$$\left.\begin{aligned}
&x_{i_{k-1}+1}^*(t) = x^*\left(t_{k-1}^+\right) + f_{q_k}\left(x^*\left(t_{k-1}^+\right), u^*\left(t_{k-1}^+\right)\right)\left((i_{k-1}+1)\Delta\tau + t - t_{k-1}\right)\\
&x_{i+1}^*(t) = x_i^*(t) + f_{q_k}\left(x_i^*(t), u_i^*(t)\right)\Delta\tau, \quad \text{for} \quad i = i_{k-1}+1, \ldots, i_k - 1\\
&x^*(t_k^-) = x_{i_k}^*(t) + f_{q_k}\left(x_{i_k}^*(t), u_{i_k}^*(t)\right)(t_k - t - i_k\Delta\tau)\\
&\psi_{q_k,q_{k+1}}\left(x^*\left(t_k^-\right)\right) = 0\\
&x^*(t_k^+) = \gamma_{q_k,q_{k+1}}\left(x^*\left(t_k^-\right)\right)
\end{aligned}\right\} \quad \text{for} \quad k = 2, \ldots, m-1, \tag{18}$$

$$x_{i_{m-1}+1}^*(t) = x^*\left(t_{m-1}^+\right) + f_{q_m}\left(x^*\left(t_{m-1}^+\right), u^*\left(t_{m-1}^+\right)\right)\left((i_{m-1}+1)\Delta\tau + t - t_{m-1}\right), \tag{19}$$

$$x_{i+1}^*(t) = x_i^*(t) + f_{q_m}\left(x_i^*(t), u_i^*(t)\right)\Delta\tau, \quad i = i_{m-1}+1, \ldots, N-1. \tag{20}$$

To derive the optimality conditions, we define the augmented cost function,[46] which is obtained by adjointing equality constraints (14)-(20) to the cost function (12). We introduce the Lagrange multipliers for the state equations in (14), (15), and (18)-(20) as $\lambda_1^*(t), \ldots, \lambda_N^*(t) \in \mathbb{R}^n$, and $\lambda^*(t_1^-), \lambda^*(t_1^+), \ldots, \lambda^*(t_{m-1}^-), \lambda^*(t_{m-1}^+) \in \mathbb{R}^n$. We also introduce the Lagrange multipliers for the switching conditions $\psi_{q_1,q_2}(x) = 0, \ldots, \psi_{q_{m-2},q_{m-1}}(x) = 0$, and $\psi_{q_{m-1},q_m}(x) = 0$ as $\nu_{q_1,q_2}^*(t) \in \mathbb{R}^{l_{q_1,q_2}}, \ldots, \nu_{q_{m-2},q_{m-1}}^*(t) \in \mathbb{R}^{l_{q_{m-2},q_{m-1}}}$ and $\nu_{q_{m-1},q_m}^*(t) \in \mathbb{R}^{l_{q_{m-1},q_m}}$. The augmented cost function $\bar{J}$ is then given as

$$\begin{aligned}
\bar{J} = J &+ \sum_{i=i_m+1}^{N-1} \lambda_{i+1}^*(t)^{\mathrm{T}}\left\{f_{q_m}\left(x_i^*(t), u_i^*(t)\right)\Delta\tau + x_i^*(t) - x_{i+1}^*(t)\right\}\\
&+ \lambda_{i_{m-1}+1}^*(t)^{\mathrm{T}}\left\{f_{q_m}\left(x^*\left(t_{m-1}^+\right), u^*\left(t_{m-1}^+\right)\right)\left((i_{m-1}+1)\Delta\tau + t - t_{m-1}\right) - x^*\left(t_{m-1}^+\right) - x_{i_{m-1}+1}^*(t)\right)\right\}\\
&+ \sum_{k=2}^{m-1}\Bigg[\nu_{q_k,q_{k+1}}^*(t)^{\mathrm{T}}\psi_{q_k,q_{k+1}}\left(x^*\left(t_k^-\right)\right) + \lambda^*\left(t_k^-\right)^{\mathrm{T}}\left\{f_{q_k}\left(x_{i_k}^*(t), u_{i_k}^*(t)\right)(t_k - t - i_k\Delta\tau) + x_{i_k}^*(t) - x^*\left(t_k^-\right)\right\}\\
&+ \sum_{i=i_{k-1}+1}^{i_k-1}\lambda_{i+1}^*(t)^{\mathrm{T}}\left\{f_{q_k}\left(x_i^*(t), u_i^*(t)\right)\Delta\tau + x_i^*(t) - x_{i+1}^*(t)\right\}\\
&+ \lambda_{i_{k-1}+1}^*(t)^{\mathrm{T}}\left\{f_{q_k}\left(x^*\left(t_{k-1}^+\right), u^*\left(t_{k-1}^+\right)\right)\left((i_{k-1}+1)\Delta\tau + t - t_{k-1}\right) + x^*\left(t_{k-1}^+\right) - x_{i_{k-1}+1}^*\right\}\Bigg]\\
&+ \nu_{q_1,q_2}^*(t)^{\mathrm{T}}\psi_{q_1,q_2}\left(x^*\left(t_1^-\right)\right) + \lambda^*\left(t_1^-\right)^{\mathrm{T}}\left\{f_{q_1}\left(x_{i_1}^*(t), u_{i_1}^*(t)\right)(t_1 - t - i_1\Delta\tau) + x_{i_1}^*(t) - x^*\left(t_1^-\right)\right\}\\
&+ \sum_{i=0}^{i_1-1}\lambda_{i+1}^*(t)^{\mathrm{T}}\left\{f_{q_1}\left(x_i^*(t), u_i^*(t)\right)\Delta\tau + x_i^*(t) - x_{i+1}^*(t)\right\}\Delta\tau.
\end{aligned} \tag{21}$$

We further introduce the Hamiltonian of subsystem $q \in Q$

$$H_q(x, u, \lambda) = L_q(x, u) + \lambda^{\mathrm{T}} f_q(x, u). \tag{22}$$

We then obtain the optimality conditions by calculus of variations[46]

$$\lambda_N^*(t) = \left(\frac{\partial\varphi_{q_m}}{\partial x}\right)^{\mathrm{T}}\left(x_N^*(t)\right), \tag{23}$$

$$\lambda_i^*(t) = \lambda_{i+1}^*(t) + \left(\frac{\partial H_{q_m}}{\partial x}\right)^{\mathrm{T}}\left(x_i^*(t), u_i^*(t), \lambda_{i+1}^*(t)\right)\Delta\tau, \quad i = i_{m-1}+1, \ldots, N-1, \tag{24}$$

$$\lambda^*\left(t_{m-1}^+\right) = \lambda^*_{i_{m-1}+1}(t) + \left(\frac{\partial H_{q_m}}{\partial x}\right)^{\mathrm{T}} \left(x^*\left(t_{m-1}^+\right), u^*\left(t_{m-1}^+\right), \lambda^*_{i_{m-1}+1}(t)\right) \left((i_{m-1}+1)\Delta\tau + t - t_{m-1}\right), \tag{25}$$

$$\left.\begin{aligned}
&\lambda^*\left(t_k^-\right) = \left(\frac{\partial \gamma_{q_k,q_{k+1}}}{\partial x}\right)^{\mathrm{T}} \left(x^*\left(t_k^-\right)\right) \lambda^*\left(t_k^+\right) + \left(\frac{\partial \psi_{q_k,q_{k+1}}}{\partial x}\right)^{\mathrm{T}} \left(x^*\left(t_k^-\right)\right) \nu^*_{q_k,q_{k+1}}(t) \\
&\lambda^*_{i_k}(t) = \lambda^*\left(t_k^-\right) + \left(\frac{\partial H_{q_k}}{\partial x}\right)^{\mathrm{T}} \left(x^*_{i_k}(t), u^*_{i_k}(t), \lambda^*\left(t_k^-\right)\right) \left(t_k - t - i_k\Delta\tau\right) \\
&\lambda^*_i(t) = \lambda^*_{i+1}(t) + \left(\frac{\partial H_{q_k}}{\partial x}\right)^{\mathrm{T}} \left(x^*_i(t), u^*_i(t), \lambda^*_{i+1}(t)\right) \Delta\tau, \quad i = i_{k-1}+1, \ldots, i_k - 1 \\
&\lambda^*\left(t_{k-1}^+\right) = \lambda^*_{i_{k-1}+1}(t) + \left(\frac{\partial H_{q_k}}{\partial x}\right)^{\mathrm{T}} \left(x^*\left(t_{k-1}^+\right), u^*\left(t_{k-1}^+\right), \lambda^*_{i_{k-1}+1}(t)\right) \left((i_{k-1}+1)\Delta\tau + t - t_{k-1}\right)
\end{aligned}\right\} \quad \text{for} \quad k = 2, \ldots, m-1, \tag{26}$$

$$\lambda^*(t_1^-) = \left(\frac{\partial \gamma_{q_1,q_2}}{\partial x}\right)^{\mathrm{T}} \left(x^*\left(t_1^-\right)\right) \lambda^*\left(t_1^+\right)\right) + \left(\frac{\partial \psi_{q_1,q_2}}{\partial x}\right)^{\mathrm{T}} \left(x^*\left(t_1^-\right)\right) \nu^*_{q_1,q_2}(t), \tag{27}$$

$$\lambda^*_{i_1}(t) = \lambda^*(\tau_1^-; t) + \left(\frac{\partial H_{q_1}}{\partial x}\right)^{\mathrm{T}} \left(x^*_{i_1}(t), u^*_{i_1}(t), \lambda^*\left(\tau_1^-; t\right)\right) (\tau_1 - i_1\Delta\tau), \tag{28}$$

$$\lambda^*_i(t) = \lambda^*_{i+1}(t) - \left(\frac{\partial H_{q_1}}{\partial x}\right)^{\mathrm{T}} \left(x^*_i(t), u^*_i(t), \lambda^*_{i+1}(t)\right) \Delta\tau, \quad i = 0, \ldots, i_1 - 1, \tag{29}$$

$$\left(\frac{\partial H_{q_m}}{\partial u}\right)^{\mathrm{T}} \left(x^*_i(t), u^*_i(t), \lambda^*_{i+1}(t)\right) = 0, \quad i = i_{m-1}+1, \ldots, N-1, \tag{30}$$

$$\left(\frac{\partial H_{q_m}}{\partial u}\right)^{\mathrm{T}} \left(x^*\left(t_{m-1}^+\right), u^*\left(t_{m-1}^+\right), \lambda^*_{i_{m-1}+1}(t)\right) = 0, \tag{31}$$

$$\left.\begin{aligned}
&\left(\frac{\partial H_{q_k}}{\partial u}\right)^{\mathrm{T}} \left(x^*_{i_k}(t), u^*_{i_k}(t), \lambda^*\left(t_k^+\right)\right) = 0 \\
&H_{q_k}\left(x^*_{i_k}(t), u^*_{i_k}(t), \lambda^*\left(t_k^-\right)\right) - H_{q_{k+1}}\left(x^*\left(t_k^+\right), u^*\left(t_k^+\right), \lambda^*_{i_k+1}(t)\right) = 0 \\
&\left(\frac{\partial H_{q_k}}{\partial u}\right)^{\mathrm{T}} \left(x^*_i(t), u^*_i(t), \lambda^*_{i+1}(t)\right) = 0, \quad i = i_{k-1}+1, \ldots, i_k - 1 \\
&\left(\frac{\partial H_{q_k}}{\partial u}\right)^{\mathrm{T}} \left(x^*\left(t_{k-1}^+\right), u^*\left(t_{k-1}^+\right), \lambda^*_{i_{k-1}+1}(t)\right) = 0,
\end{aligned}\right\} \quad \text{for} \quad k = 2, \ldots, m-1, \tag{32}$$

$$\left(\frac{\partial H_{q_1}}{\partial u}\right)^{\mathrm{T}} \left(x^*_{i_1}(t), u^*_{i_1}(t), \lambda^*\left(t_1^+\right)\right) = 0, \tag{33}$$

$$H_{q_1}\left(x^*_{i_1}(t), u^*_{i_1}(t), \lambda^*\left(t_1^-\right)\right) - H_{q_2}\left(x^*\left(t_1^+\right), u^*\left(t_1^+\right), \lambda^*_{i_1+1}(t)\right) = 0, \tag{34}$$

$$\left(\frac{\partial H_{q_1}}{\partial u}\right)^{\mathrm{T}} \left(x^*_i(t), u^*_i(t), \lambda^*_{i+1}(t)\right) = 0, \quad i = 0, \ldots, i_1 - 1, \tag{35}$$

The unknown quantities of the optimal control problem are $x_0^*(t), \ldots, x_N^*(t)$, $x^*(t_1^-), \ldots, x^*(t_{m-1}^-)$, $x^*(t_1^+), \ldots, x^*(t_{m-1}^+)$, $u_0^*(t), \ldots, u_{N-1}^*(t)$, $u^*(t_1^+), \ldots, u^*(t_{m-1}^+)$, $\lambda_1^*(t), \ldots, \lambda_N^*(t)$, $\lambda^*(t_1^-), \ldots, \lambda^*(t_{m-1}^-)$, $\lambda^*(t_1^+), \ldots, \lambda^*(t_{m-1}^+)$, and $t_1, t_2, \ldots, t_m$. For a given $u_0^*(t), \ldots, u_{N-1}^*(t), u^*(t_1^+), \ldots, u^*(t_{m-1}^+)$, and $t_1, \ldots, t_{m-1}$, we can specify $i_1, \ldots, i_{m-1}$ and compute $x_0^*(t), \ldots, x_N^*(t)$, $x^*(t_1^-), \ldots, x^*(t_{m-1}^-), x^*(t_1^+), \ldots, x^*(t_{m-1}^+)$ from (13)-(20), and $\lambda_1^*(t), \ldots, \lambda_N^*(t)$, $\lambda^*(t_1^-), \ldots, \lambda^*(t_{m-1}^-)$, $\lambda^*(t_1^+), \ldots, \lambda^*(t_{m-1}^+)$ from (23)-(29). Thus, we define variables to be determined as

$$U(t) := \begin{bmatrix} u_0^*(t) \\ \vdots \\ u_{N-1}^*(t) \\ U_{q_1,q_2}(t) \\ \vdots \\ U_{q_{m-1},q_m}(t) \end{bmatrix}, \quad U_{q_k,q_{k+1}}(t) := \begin{bmatrix} u^*\left(t_k^+\right) \\ \nu^*_{q_k,q_{k+1}}(t) \\ t_k \end{bmatrix}, \tag{36}$$

satisfying

$$F(U(t),x(t),t) := \begin{bmatrix} \left(\frac{\partial H_{q_1}}{\partial u}\right)^{\mathrm{T}} \left(x_0^*(t),u_0^*(t),\lambda_1^*(t)\right) \\ \vdots \\ \left(\frac{\partial H_{q_{k+1}}}{\partial u}\right)^{\mathrm{T}} \left(x_{N-1}^*(t),u_{N-1}^*(t),\lambda_N^*(t)\right) \\ F_{q_1,q_2}(U(t),x(t),t) \\ \vdots \\ F_{q_{m-1},q_m}(U(t),x(t),t) \end{bmatrix} = 0, \tag{37}$$

where $F_{q_k,q_{k+1}}(U(t),x(t),t)$ is

$$F_{q_k,q_{k+1}}(U(t),x(t),t) := \begin{bmatrix} \left(\frac{\partial H_{q_{k+1}}}{\partial u}\right)^{\mathrm{T}} \left(x^*\left(t_k^+\right),u^*\left(t_k^+\right),\lambda_{i_k+1}^*(t)\right) \\ \psi_{q_k,q_{k+1}}\left(x^*\left(t_k^-\right)\right) \\ H_{q_k}\left(x_{i_k}^*(t),u_{i_k}^*(t),\lambda^*\left(t_k^-\right)\right) - H_{q_{k+1}}\left(x^*\left(t_k^+\right),u^*\left(t_k^+\right),\lambda_{i_k+1}^*(t)\right) \end{bmatrix}, \tag{38}$$

for $k = 1, \ldots, m-1$. Note that, if $\sigma = (q)$, ie, if only one subsystem q is active over the whole horizon, the optimal control problem is just given as the conventional one for subsystem q.

4 | THE MOVING SWITCHING SEQUENCE APPROACH

The moving switching sequence approach considers the recession of the switching sequence on the horizon as well as the optimal state trajectory and the optimal control input on the horizon. To justify this approach, we make the following assumption.

Assumption 6. Let $\sigma = (q_1, \ldots, q_m)$ be the switching sequence on the horizon at the sampling time t. The switching sequence at the next sampling time $t + \Delta t$ is given as one of the following: $\sigma = (q_1, \ldots, q_m)$, $\sigma = (q_1, \ldots, q_m, q_{m+1})$ for some $q_{m+1} \in Q$, $\sigma = (q_2, \ldots, q_m)$, or $\sigma = (q_2, \ldots, q_m, q_{m+1})$.

Figure 3 represents possible changes of the switching sequence under this assumption. This assumption limits the change in the switching sequence to those only by the exit of the first switch of the switching sequence because of an actual switch occurrence or the addition of a new switch to the end of the switching sequence. This is a reasonable assumption for systems with state-dependent switches as long as the actual state changes continuously. If the state changes continuously, ie, the actual state changes slightly at each time in a sufficiently short sampling period, the optimal state trajectory also changes slightly from that of the previous sampling time. This assumption will then hold as long as the distance between each switching condition is sufficiently far.

Under this assumption, the switching sequence $\sigma = (q_1, \ldots, q_m)$ moves to the present time at each time without changing from the previous time to the current time and an additional switch may occur at $t' \in (t_m, t+T]$, where the last subsystem of σ is active. We update the switching sequence by checking whether an additional switch occurs at $t' \in (t_m, t+T]$ at each time and whether the actual switch occurs or not between the current sampling time and the next sampling time. We check the former by computing the state trajectory on the horizon on the basis of the current solution and the current switching sequence. If we detect an additional switch on the state trajectory, we reconstruct the solution taking the switch into account. We check the latter after the solution is updated by evaluating the updated switching instants. If the instant of the first switch of the switching sequence after the solution is updated is less than the next sampling time, we predict the actual switch between the current sampling time and the next sampling time and then remove variables related to the switch from the solution. We propose an algorithm of the moving switching sequence approach composed of two main parts: (1) the optimization of the control input and switching instants under the switching sequence obtained at the previous sampling time by solving the TPBVP formulated in Section 3 and (2) the reconstruction of the solution by solving the reduced optimal control problem for the additional switch. For (1), we utilize the C/GMRES method, and for (2), we define and solve the optimal control problem just for variables with respect to the additional switch and additional active subsystem and reconstruct the solution.

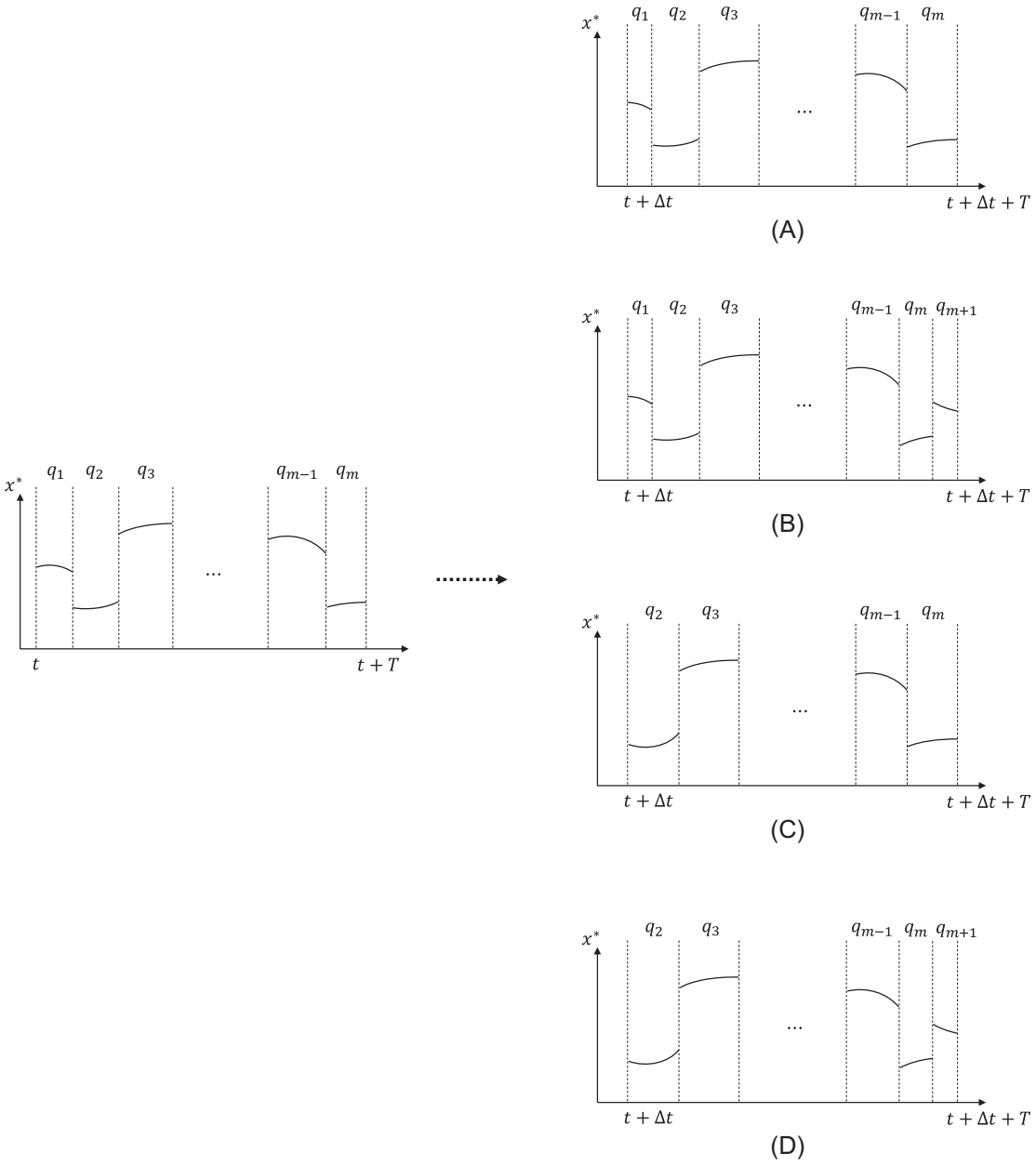

FIGURE 3 Possible changes of the switching sequence σ from t to $t + \Delta t$ under Assumption 6. A, The switching sequence dows not change; B, An additional switch from subsystem q_m to subsystem q_{m+1} is detected in the last interval of the switching sequence; C, An actual switching from subsystem q_1 to subsystem q_2 occurs between t and $t + \Delta t$; D, An actual switch from subsystem q_1 to subsystem q_2 occurs between t and $t + \Delta t$, and an additional switch from subsystem q_m to subsystem q_{m+1} is detected in the last interval of the switching sequence

In Section 4.1, we briefly introduce the C/GMRES method for the TPBVP and the method to update the solution in the C/GMRES method in accordance with the switches. In Section 4.2, we describe the detection of an additional switch and optimization of variables related to an additional switch. In Section 4.3, we summarize the algorithm of the moving switching sequence algorithm.

4.1 | The C/GMRES method and update of the solution

The C/GMRES method

To realize NMPC for systems with state-dependent switches, we have to compute the unknown quantities to be determined, $U(t)$, which is defined by (36) and satisfies (37), at each sampling time within a given sampling period. The C/GMRES method[37] is a fast algorithm of NMPC composed of the continuation method[47] and the GMRES method.[48] In

the C/GMRES method, we do not compute $U(t)$ directly but update the solution by integrating $\dot{U}(t)$ at each time, eg, as

$$U(t + \Delta t) = U(t) + \dot{U}(t)\Delta t, \tag{39}$$

where Δt is the sampling period, and the nonlinear equation $F(U(t), x(t), t) = 0$ for $U(t)$ is transformed into a linear equation for $\dot{U}(t)$ by the continuation method as

$$\frac{\partial F}{\partial U}\dot{U} = -\zeta F - \frac{\partial F}{\partial x}\dot{x} - \frac{\partial F}{\partial t}, \tag{40}$$

where $\zeta > 0$ is a stabilization parameter. We solve this equation by using the forward-difference GMRES method (FD-GMRES method),[48] which solves linear equations fast. In the FD-GMRES method, forward-difference approximation is used to approximate the products of Jacobians and some $W \in \mathbb{R}^{n'}$, $w \in \mathbb{R}^n$, $\omega \in \mathbb{R}$ as

$$\frac{\partial F}{\partial U}W + \frac{\partial F}{\partial x}w + \frac{\partial F}{\partial t}\omega \simeq \frac{F(U + hW, x + hw, t + h\omega) - F(U, x, t)}{h}, \tag{41}$$

where $h > 0$ is a time increment. By utilizing the FD-GMRES method for the TPBVP formulated in Section 3, we obtain not only the time derivatives of the optimal control input sequence but also the time derivatives of the optimal switching instants and the corresponding Lagrange multipliers with respect to the switching conditions.

Integration of $\dot{U}(t)$ taking discontinuity of the control input into account

The integration (39) is based on the continuity of $U(t)$ between t and $t + \Delta t$, and we have to modify the numerical integration method when there are switches and state jumps because that continuity does not hold when there are switches or state jumps. Let $\sigma = (q_1, \ldots, q_m)$ be the switching sequence for the NOCP at time t, let $t_k(t)$ be the optimal switching instants from subsystem q_k to subsystem q_{k+1} at time t for $k = 1, \ldots, m-1$, and let $i_k(t)$ be an integer satisfying $i_k(t)\Delta\tau \leq t_k(t) - t < (i_k(t) + 1)\Delta\tau$. Note that subsystem q_k is active at $t' \in (t_k(t), t_{k+1}(t))$ in the NOCP at time t, and then, $u^*(t_k^+(t))$, $u^*_{i_k(t)+1}(t)$, $\ldots$, $u^*_{i_{k+1}(t)-1}(t)$, and $u^*_{i_{k+1}(t)}(t)$ are the optimal control input on the interval where subsystem q_k is active.

Figure 4 illustrates the update of the control input taking the discontinuity of the control input into account. Black circles represent $u_i^*(t)$ and $u_i^*(t + \Delta t)$ for $i = i_k(t), \ldots, i_{k+1}(t) + 1$, and white circles represent $u^*(t_k^+(t))$, $u^*(t_{k+1}^+(t))$, $u^*(t_k^+(t + \Delta t))$, and $u^*(t_{k+1}^+(t + \Delta t))$. We then have to determine the control input for subsystem q_k at time $t + \Delta t$ on the basis of control inputs and their time derivatives for subsystem q_k at time t, ie, we have to determine $u^*(t_k^+(t + \Delta t))$, $u^*_{i_k(t+\Delta t)+1}(t + \Delta t)$, $\ldots$, $u^*_{i_{k+1}(t+\Delta t)-1}(t + \Delta t)$, and $u^*_{i_{k+1}(t+\Delta t)}(t + \Delta t)$ on the basis of $u^*(t_k^+(t))$, $u^*_{i_k(t)+1}(t)$, $\ldots$, $u^*_{i_{k+1}(t)}(t)$, $\dot{u}^*(t_k^+(t))$, $\dot{u}^*_{i_k(t)+1}(t)$, $\ldots$, $\dot{u}^*_{i_{k+1}(t)-1}(t)$, and $\dot{u}^*_{i_{k+1}(t)}(t)$. However, when $i_k(t) \neq i_k(t+\Delta t)$, it is not suitable to determine $u_i^*(t+\Delta t)$ by (39) for i such that $i_k(t + \Delta t) < i \leq i_k(t)$ or $i_k(t) < i \leq i_k(t + \Delta t)$ because (39) does not consider the difference between the active subsystem at $i\Delta\tau$ at the sampling time t and that of $i\Delta\tau$ at the sampling time $t + \Delta t$. For example, in Figure 4, ie, when $i_k(t) = i_k(t + \Delta t) + 1$ and $i_{k+1}(t) = i_{k+1}(t + \Delta t) - 1$, it is not suitable to determine $u^*_{i_k(t)}(t + \Delta t)$ by (39) because control input $u^*_{i_k(t)}(t)$ and its time derivative $\dot{u}^*_{i_k(t)}(t)$ are for subsystem q_{k-1} whereas $u^*_{i_k(t+\Delta t)}(t + \Delta t)$ is for subsystem q_k. Additionally, it is also not suitable to determine $u^*_{i_{k+1}(t)}(t + \Delta t)$ by (39) because control input $u^*_{i_{k+1}(t)+1}(t)$ and its time derivative $\dot{u}^*_{i_{k+1}(t)+1}(t)$ are for q_{k+1} whereas $u^*_{i_{k+1}(t+\Delta t)+1}(t + \Delta t)$ is for q_k. The update (39) when $i_k(t) \neq i_k(t + \Delta t)$ is modified as follows. We first compute $u^*(t_k^+(t + \Delta t))$ for $k = 1, \ldots, m-1$ by

$$u^*\left(t_k^+(t + \Delta t)\right) = u^*\left(t_k^+(t)\right) + \dot{u}^*\left(t_k^+(t)\right)\Delta t, \tag{42}$$

and the switching instants $t_k(t + \Delta t)$ for $k = 1, \ldots, m-1$ by

$$t_k(t + \Delta t) = t_k(t) + \dot{t}_k(t)\Delta t. \tag{43}$$

Next, we compute $u_i^*(t + \Delta t)$ for i satisfying $i_k(t) < i \leq i_{k+1}(t)$ and $i_k(t + \Delta t) < i \leq i_{k+1}(t + \Delta t)$ by (39), ie, by

$$u^*_{i+1}(t) = u^*_i(t) + \dot{u}^*_i(t)\Delta t. \tag{44}$$

Thin arrows in Figure 4 represent updates by (42) and (44). We update $u_i^*(t + \Delta t)$ for i such that $i_k(t + \Delta t) < i \leq i_k(t)$ or $i_k(t) < i \leq i_k(t + \Delta t)$ using the control input for q_k at the sampling time $t + \Delta t$ and its time derivative. We regard $u_i^*(t + \Delta t)$

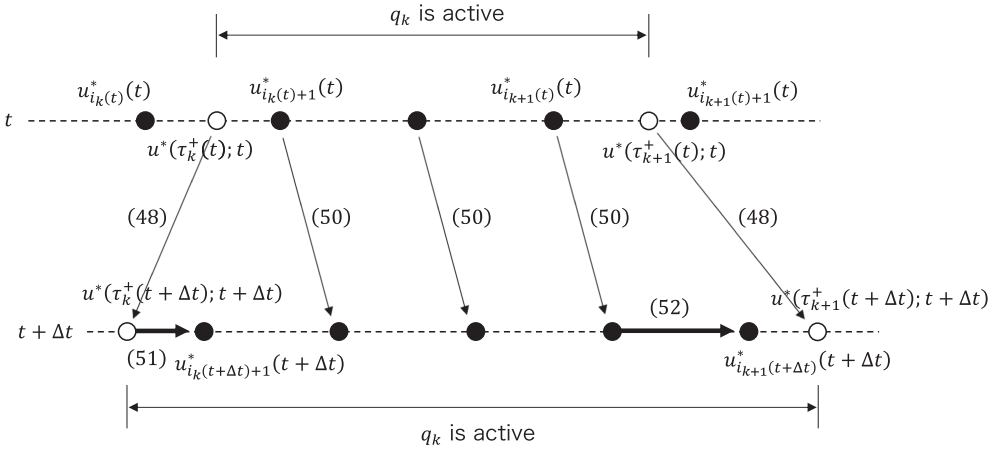

FIGURE 4 Update of the control input in consideration of switches and state jumps

for i such that $i_k(t+\Delta t) < i \leq i_k(t)$ as the control input obtained by integrating $\dot{u}^*(t^+_{k-1}(t))$ for $i\Delta\tau + t - t_{k-1}(t+\Delta t)$ from $u^*(t^+_k(t))$, ie, we determine $u^*_i(t+\Delta t)$ for $i_k(t+\Delta t) < i \leq i_k(t)$ as

$$u^*_i(t+\Delta t) = u^*\left(t^+_{k-1}(t+\Delta t)\right) + \dot{u}^*\left(t^+_{k-1}(t)\right)\left(i\Delta\tau + t - t_{k-1}(t+\Delta t)\right). \tag{45}$$

Additionally, we regard $u^*_i(t+\Delta t)$ for i such that $i_k(t) < i \leq i_k(t+\Delta t)$ as the control input obtained by integrating $\dot{u}^*_{i_k(t)}(t)$ for $(i - i_k(t))\Delta\tau$ from $u^*_{i_k(t)}(t+\Delta t)$ as

$$u^*_i(t+\Delta t) = u^*_{i_k(t)}(t+\Delta t) + \dot{u}^*_{i_k(t)}(t)(i - i_k(t))\Delta\tau. \tag{46}$$

Thick arrows in Figure 4 represent the integration by (45) and (46). Note that the Lagrange multipliers for the switching conditions $\nu^*_{q_1,q_2}(t+\Delta t), \ldots, \nu^*_{q_{m-1},q_m}(t+\Delta t)$ are just updated by (39), ie, as

$$\nu^*_{q_k,q_{k+1}}(t+\Delta t) = \nu^*_{q_k,q_{k+1}}(t) + \dot{\nu}^*_{q_k,q_{k+1}}(t)\Delta t \tag{47}$$

for $k = 1, \ldots, m-1$. Note that all time derivatives in (42)-(47) are determined as components of $\dot{U}(t)$.

Prediction of the actual switch

Suppose that the switching sequence is given as $\sigma = (q_1, q_2, \ldots, q_m)$. If the switching instant $t_1(t+\Delta t)$ in $U(t+\Delta t)$ after the update of $U(t)$ by the C/GMRES method satisfies $t_1(t+\Delta t) < t + \Delta t$, we predict that the switch from subsystem q_1 to q_2 occurs on the actual system between t and $t+\Delta t$. We then remove q_1 from σ and $U_{q_1,q_2}(t+\Delta t)$ from $U(t+\Delta t)$. Note that $u^*_0(t+\Delta t)$ is already updated for subsystem q_2 by (45) in this case because $i_1(t+\Delta t) < 0 \leq i_1(t)$ holds from the fact that $t_1(t+\Delta t) < t + \Delta t$.

4.2 | Detection of an additional switch and reconstruction of the solution

Suppose that the switching sequence is given as $\sigma = (q_1, q_2, \ldots, q_m)$. In solving the TPBVP, the state trajectory on the horizon is computed on the basis of $x(t)$, σ, and $U(t)$, and we then check whether an additional switch occurs or not for $t_{m-1} < t' \leq t + T$. In the discretized problem, the state trajectory for $t_{m-1} < t' \leq t + T$ is given as $x^*(t^+_{m-1}(t)), x^*_{i_{m-1}+1}(t), \ldots, x^*_N(t)$, which is computed by (19) and (20), and we check whether an additional switch occurs or not between t_{m-1} and $(i_{m-1}+1)\Delta\tau + t$ and between $i\Delta\tau + t$ and $(i+1)\Delta\tau + t$ for $i = i_{m-1}+1, \ldots, N-1$ by evaluating $\psi_{q_m,q_k}(x)$ for all $q_k \in Q$. For example, when $l_{q_m,q_k} = 1$ for all $q_k \in Q$, we regard that a switch from subsystem q_m to q_k occurs when the sign of $\psi_{q_m,q_k}(x^*(t^+_m(t)))$ and that of $\psi_{q_m,q_k}(x^*_{i_m}(t))$ are different or when that of $\psi_{q_m,q_k}(x^*_i(t))$ and that of $\psi_{q_m,q_k}(x^*_{i+1}(t))$ are different for some $i = i_{m-1}+1, \ldots, N-1$ because $\psi_{q_m,q_k}(x)$ is continuous for $x \in \chi_{q_m}$.

When we detect an additional switch in computing the state trajectory, we reconstruct the solution $U(t)$ taking the additional switch and the additional active subsystem into account. Suppose that we detect a switch from q_m to q_{m+1} between $i_m\Delta\tau + t$ and $(i_m+1)\Delta\tau + t$ where $i_{m-1} < i_m < N$. Note that, if the switch is detected between t_{m-1} and $(i_{m-1}+1)\Delta\tau + t$, we can formulate the NOCP for reconstructing the solution just by replacing $x^*_{i_m}(t)$, $u^*_{i_m}(t)$, and $i_m\Delta\tau + t$ in the following formulation with $x^*(\tau^+_{m-1}(t))$, $u^*(\tau^+_{m-1}(t))$, and t_{m-1}, respectively. When we detect a switch from q_m to q_{m+1} between $i_m\Delta\tau + t$ and $(i_m+1)\Delta\tau + t$, then $t_m(t)$, $\nu^*_{q_m,q_{m+1}}(t)$, and $u^*(t^+_m(t))$ are appended to $U(t)$. In addition, we have

to reoptimize $u^*_{i_m+1}(t), \ldots, u^*_{N-1}(t)$ for the additional active subsystem q_{m+1} because $u^*_{i_m+1}(t), \ldots, u^*_{N-1}(t)$ are the optimal control input sequence for subsystem q_m before detecting the additional switch. To reduce the computational cost of the reconstruction of the solution, we define an optimal control problem only for the interval after the additional switch to determine $t_m(t), \nu^*_{q_m,q_{m+1}}(t), u^*(t_m^+(t))$, and $u^*_{i_m+1}(t), \ldots, u^*_{N-1}(t)$ numerically. Although this optimal control problem only considers the optimality of the interval where q_{m+1} is active, we resolve the entire optimal control problem in consideration of the whole horizon after solving this optimal control problem. We define the reduced optimal control problem for these variables as follows: find the optimal control input $u^*(t_m^+(t)), u^*_{i_m}(t), \ldots, u^*_{N-1}(t)$ minimizing the cost function defined for $t' \in (i_m\Delta\tau + t, t+T]$,

$$\begin{aligned} J_{m+1} = {} & \varphi_{q_{m+1}}\left(x^*_N(t)\right) + \sum_{i=i_m+1}^{N-1} L_{q_{m+1}}\left(x^*_i(t), u^*_i(t)\right)\Delta\tau \\ & + L_{q_{m+1}}\left(x^*\left(t_m^+(t)\right), u^*\left(t_m^+(t)\right)\right)((i_m+1)\Delta\tau + t - t_m) + L_{q_m}\left(x^*_{i_m-1}(t), u^*_{i_m-1}(t)\right)(t_m - t - i_m\Delta\tau), \end{aligned} \tag{48}$$

subject to the state trajectory for $t' \in (t + i_m\Delta\tau, t+T]$,

$$x^*\left(t_m^-(t)\right) = x^*_{i_m}(t) + f_{q_m}\left(x^*_{i_m}(t), u^*_{i_m}(t)\right)(t_m - t - i_m\Delta\tau), \tag{49}$$

$$\psi_{q_m,q_{m+1}}\left(x^*\left(t_m^-(t)\right)\right) = 0, \tag{50}$$

$$x^*\left(t_m^+(t)\right) = \gamma_{q_m,q_{m+1}}(x^*(t_m^-(t))), \tag{51}$$

$$x^*_{i_m+1}(t) = x^*\left(t_m^+(t)\right) + f_{q_{m+1}}\left(x^*\left(t_m^+(t)\right), u^*\left(t_m^+(t)\right)\right)((i_m+1)\Delta\tau + t - t_m), \tag{52}$$

$$x^*_{i+1}(t) = x^*_i(t) + f_{q_{m+1}}\left(x^*_i(t), u^*_i(t)\right)\Delta\tau,\ i = i_m+1, \ldots, N-1. \tag{53}$$

The optimality conditions for this optimal control problem are derived as

$$\lambda^*_N(t) = \left(\frac{\partial\varphi_{q_{m+1}}}{\partial x}\right)^{\mathrm{T}}\left(x^*_N(t)\right), \tag{54}$$

$$\lambda^*_i(t) = \lambda^*_{i+1}(t) + \left(\frac{\partial H_{q_{m+1}}}{\partial x}\right)^{\mathrm{T}}\left(x^*_i(t), u^*_i(t), \lambda^*_{i+1}(t)\right)\Delta\tau,\ i = i_m+1, \ldots, N-1, \tag{55}$$

$$\lambda^*\left(t_m^+(t)\right) = \lambda^*_{i_m+1}(t) + \left(\frac{\partial H_{q_{m+1}}}{\partial x}\right)^{\mathrm{T}}\left(x^*\left(t_m^+(t)\right), u^*\left(t_m^+(t)\right), \lambda^*_{i_m+1}(t)\right)((i_m+1)\Delta\tau + t - t_m), \tag{56}$$

$$\lambda^*(t_m^-(t)) = \left(\frac{\partial\gamma_{q_m,q_{m+1}}}{\partial x}\right)^{\mathrm{T}}(x^*(t_m^-(t)))\,\lambda^*\left(t_m^+(t)\right) + \left(\frac{\partial\psi_{q_m,q_{m+1}}}{\partial x}\right)^{\mathrm{T}}(x^*(t_m^-(t)))\,\nu^*_{q_m,q_{m+1}}(t), \tag{57}$$

$$\left(\frac{\partial H_{q_{m+1}}}{\partial u}\right)^{\mathrm{T}}\left(x^*_i(t), u^*_i(t), \lambda^*_{i+1}(t)\right) = 0,\ i = i_m+1, \ldots, N-1, \tag{58}$$

$$\left(\frac{\partial H_{q_{m+1}}}{\partial u}\right)^{\mathrm{T}}\left(x^*\left(t_m^+(t)\right), u^*\left(t_m^+(t)\right), \lambda^*_{i_m+1}(t)\right) = 0, \tag{59}$$

$$H_{q_m}\left(x^*_{i_m}(t), u^*_{i_m}(t), \lambda^*(t_m^-(t))\right) - H_{q_{m+1}}\left(x^*\left(t_m^+(t)\right), u^*\left(t_m^+(t)\right), \lambda^*_{i_m+1}(t)\right) = 0. \tag{60}$$

The unknown quantities in this optimal control problem are given as $x^*_{i_m}(t), x^*(t_m^-(t)), x^*(t_m^+(t)), x^*_{i_m+1}(t), \ldots, x^*_N(t), u^*_{i_m}(t)$, $u^*(t_m^+(t)), u^*_{i_m+1}(t), \ldots, u^*_{N-1}(t), \lambda^*(t_m^-(t)), \lambda^*(t_m^+(t)), \lambda^*_{i_m+1}(t), \ldots, \lambda^*_N(t), t_m$, and $\nu^*_{q_m,q_{m+1}}(t)$. Note that $u^*_{i_m}(t)$ and $x^*_{i_m}(t)$ are given as the boundary value. For a given $u^*(t_m^+(t)), u^*_{i_m}(t), \ldots, u^*_{N-1}(t), t_m(t)$, and $\nu^*_{q_m,q_{m+1}}(t)$, we can compute $x^*(t_m^-(t)), x^*(t_m^+(t))$, $\ldots, x^*_N(t)$ from (49) and (51)-(53) and $\lambda^*_N(t), \ldots, \lambda^*(t_m^+(t))$ from (54)-(57). Therefore, variables to be determined for this optimal control problem are given as

$$U_{q_{m+1}}(t) := \begin{bmatrix} u^*\left(t_m^+(t)\right) \\ u^*_{i_m+1}(t) \\ \vdots \\ u^*_{N-1}(t) \\ \nu^*_{q_m,q_{m+1}}(t) \\ t_m(t) \end{bmatrix}, \tag{61}$$

and $U_{q_{m+1}}(t)$ has to satisfy

$$F_{q_{m+1}}(U_{q_{m+1}}(t)) := \begin{bmatrix} \left(\frac{\partial H_{q_{m+1}}}{\partial u}\right)^{\mathrm{T}} \left(x^*\left(t_m^+(t)\right), u^*\left(t_m^+(t)\right), \lambda^*_{i_m+1}(t)\right) \\ \left(\frac{\partial H_{q_{m+1}}}{\partial u}\right)^{\mathrm{T}} \left(x^*_{i_m+1}(t), u^*_{i_m+1}(t), \lambda^*_{i_m+2}(t)\right) \\ \vdots \\ \left(\frac{\partial H_{q_{m+1}}}{\partial u}\right)^{\mathrm{T}} \left(x^*_{N-1}(t), u^*_{N-1}(t), \lambda^*_N(t)\right) \\ \psi_{q_m,q_{m+1}}\left(x^*\left(t_m^+(t)\right)\right) \\ H_{q_{m+1}}\left(x^*_{i_m}(t), u^*_{i_m}(t), \lambda^*\left(t_m^-(t)\right)\right) - H_{q_{m+1}}\left(x^*(t_m^+(t)), u^*(t_m^+(t)), \lambda^*_{i_m+1}(t)\right) \end{bmatrix} = 0. \tag{62}$$

We solve this problem numerically using the conventional methods, eg, Newton's method. For computation using Newton's method, we make the following assumption for $F_{q_{m+1}}(U_{q_{m+1}}(t))$ and $U_{q_{m+1}}(t)$.

Assumption 7. The Jacobian matrix of $F_{q_{m+1}}(U_{q_{m+1}}(t))$ with respect to $U_{q_{m+1}}(t)$ is nonsingular for all $U_{q_{m+1}}(t) \in \mathbb{R}^{(N-i_m)m+l_{q_m,q_{m+1}}+1}$.

Note that, if Assumption 7 does not hold, we add a minimum number of the control inputs $u^*_{i_m-k}(t), \ldots, u^*_{i_m-1}(t), u^*_{i_m}(t)$ to $U_{q_{m+1}}(t)$ so that the Jacobian of $F_{q_{m+1}}(U_{q_{m+1}}(t))$ for $U_{q_{m+1}}(t)$ becomes nonsingular and redefine the optimal control problem.

Algorithm 1 shows the reconstruction of the solution for an additional switch. First, if we detect a switch between i_m and i_m+1, where $i_m < N-1$, we set $t_m(t)$ between $i_m\Delta\tau + t$ and $(i_m+1)\Delta\tau + t$. For example, it is reasonable to approximate $t_m(t)$ by linear interpolation as

$$t_m(t) = \frac{\left\|\psi_{q_m,q_{m+1}}\left(x^*_{i_m}(t)\right)\right\|}{\left\|\psi_{q_m,q_{m+1}}\left(x^*_{i_m}(t)\right)\right\| + \left\|\psi_{q_m,q_{m+1}}\left(x^*_{i_m+1}(t)\right)\right\|}\Delta\tau + i_m\Delta\tau + t. \tag{63}$$

Next, we substitute initial guess values for $u^*(t_m^+(t)), u^*_{i_m}(t), \ldots, u^*_{N-1}(t)$, and $v^*_{q_m,q_{m+1}}(t)$. After that, we iterate Newton's method for (62) and obtain $U_{q_{m+1}}(t)$. We terminate the iteration when the error in the optimality condition $\|F_{q_{m+1}}(U_{q_{m+1}}(t))\|$ is sufficiently small or when the number of iteration is sufficiently large. We introduce $\tilde{r} \in \mathbb{R}$ for the criterion of the former and $i_{\max} \in \mathbb{N}$ for that of the latter. After the iteration, we replace $u^*_{i_m+1}(t), \ldots, u^*_{N-1}(t)$ in $U(t)$ with $u^*_{i_m+1}(t), \ldots, u^*_{N-1}(t)$ in $U_{q_{m+1}}(t)$ and append

$$U_{q_m,q_{m+1}}(t) := \begin{bmatrix} u^*\left(t_m^+(t)\right) \\ v^*_{q_m,q_{m+1}}(t) \\ t_m(t) \end{bmatrix} \tag{64}$$

to $U(t)$, and append q_{m+1} to σ.

Algorithm 1 Reconstruction of $U(t)$ for an additional switch

Require: $i_m, x^*_{i_m}(t)$, and $u^*_{i_m}(t)$

1: Set $t_m(t)$ s.t. $t + (i_m - 1)\Delta\tau < t_m(t) \le i_m\Delta\tau$, eg, by (63).
2: Substitute initial guess values to $u^*(t_m(t)), u^*_{i_m}(t), \ldots, u^*_{N-1}(t)$, and $v^*_{q_m,q_{m+1}}(t)$.
3: Compute $x^*(t_m^-(t)), x^*(t_m^+(t)), x^*_{i_m+1}(t), x^*_N(t), \lambda^*_N(t), \ldots, \lambda^*_{i_m+1}(t), \lambda^*(t_m^+(t)), \lambda^*(t_m^-(t))$, and compute $F_{q_{m+1}}(U_{q_{m+1}}(t)) = 0$.
4: **while** $|F_{q_{m+1}}(U_{q_{m+1}}(t))| < r_{\text{init}}$ or $i < i_{\max}$ **do**
5: Obtain $\Delta U_{q_{m+1}}$ by Newton's method for $F_{q_{m+1}}(U_{q_{m+1}}(t))$
6: Update $U_{q_{m+1}}(t)$ as $U_{q_{m+1}}(t) := U_{q_{m+1}}(t) + \Delta U_{q_{m+1}}$
7: Compute $x^*(t_m^-(t)), x^*(t_m^+(t)), x^*_{i_m+1}(t), \ldots, x^*_N(t), \lambda^*_N(t), \ldots, \lambda^*_{i_m+1}(t), \lambda^*(t_m^+(t))$, and $F_{q_{m+1}}(U_{q_{m+1}}(t))$
8: **end while**
9: Replace $u^*_{i_m+1}(t), \ldots, u^*_{N-1}(t)$ in $U(t)$ with $u^*_{i_m+1}(t), \ldots, u^*_{N-1}(t)$ in $U_{q_{m+1}}(t)$ and append $U_{q_m,q_{m+1}}(t)$ to $U(t)$ and q_{m+1} to σ.

4.3 | The algorithm of the moving switching sequence approach

The length of the horizon

We set the length of the horizon as a smooth increasing function such that $T(0) = 0$ and $T(t) \to T_f (t \to \infty)$, eg, as

$$T(t) = T_f(1 - e^{-\alpha t}), \tag{65}$$

where $T_f \in \mathbb{R}$ and $\alpha \in \mathbb{R}$ are positive constant values, to initialize $U(0)$ in sufficiently short computational time and to modify the solution $U(t)$ when Assumption 6 does not hold.

Initialization of $U(0)$

We have to compute the initial solution $U(0)$ by iterative method such as Newton's method instead of the C/GMRES method because we do not know the optimal $U(t)$ before $t = 0$. We also have to compute $U(0)$ in a short computational time unless we know $x(0)$ and compute $U(0)$ in advance. If we set $T(t)$ such that $T(0) = 0$, we can compute the initial solution $U(0)$ in sufficiently short computational time even by iterative method such as Newton's method: from the initial state $x(0)$ and active subsystem q at $t = 0$, we obtain $u(0) \in \mathbb{R}^m$ by solving

$$\left(\frac{\partial H_q}{\partial u}\right)^{\mathrm{T}}(x(0), u(0), \lambda_0) = 0, \tag{66}$$

where $\lambda_0 \in \mathbb{R}^n$ is obtained by

$$\lambda_0 = \left(\frac{\partial \varphi_q}{\partial x}\right)^{\mathrm{T}}(x(0)). \tag{67}$$

and we set $u_i^*(0) = u(0)$ for all $i = 0, \dots, N-1$. Note that $U(t)$ obtained by this initialization satisfies $F(U(t), x(t), t) = 0$. Note also that, if we set the sampling period sufficiently short, $T(t)$ and $T(t + \Delta t)$ are almost same and we can update $U(t)$ just by integrating $\dot{U}(t)$.

Shrinkage of the horizon when Assumption 6 does not hold

When we detect an additional switch before the end of the switching sequence in computing the state on the horizon, ie, when Assumption 6 does not hold, we can still use the C/GMRES method by shrinking the horizon and then increasing the length of the horizon smoothly. Suppose that $\sigma = (q_1, q_2, \dots, q_m)$ is the switching sequence at time t and an additional switch is detected at the middle of the switching sequence, eg, an additional switch from q_k to q_j ($1 \leq j < m, q_j \neq q_{k+1}$) is detected between i_j and $i_j + 1$ ($i_k + 1 \leq i_j < i_{k+1}$). We then shrink the length of the horizon $T(t)$ by setting $t_0 > 0$ so that $T(t)$ satisfies

$$T(t) := T_f(1 - e^{-\alpha(t - t_0)}) \leq i_j \Delta\tau - t. \tag{68}$$

We also modify $u_0^*(t), \dots, u_{N-1}^*(t)$ by linear interpolation of $u_0^*(t), \dots, u_{i_j}^*(t)$. Let $\Delta\tau_1 := T_f(1 - e^{-\alpha t})/N$ and $\Delta\tau_2 := T_f(1 - e^{-\alpha(t-t_0)})/N$. Then, we approximate $u_0^*(t), u_1^*(t), \dots, u_{N-1}^*(t)$, which correspond to $u^*(t), u^*(t+\Delta\tau_2), \dots, u^*(t+(N-1)\Delta\tau_2)$, by linear interpolation from $u_0^*(t), \dots, u_{N-1}^*(t)$ before the modification, which correspond to $u^*(t), u^*(t+\Delta\tau_1), \dots, u^*(t+i_j\Delta\tau_1)$. Figure 5 shows an example of the linear interpolation of $u_0^*(t), \dots, u_{N-1}^*(t)$ when we shrink the horizon. After interpolating $u_0^*(t), \dots, u_{N-1}^*(t)$, we remove $U_{q_k,q_{k+1}}(t), \dots, U_{q_{m-1},q_m}(t)$ from $U(t)$ and remove $q_{k+1}, \dots, q_m$ from σ. Once the solution $U(t)$ for the shrunk horizon is obtained, we can continue to update the solution by the C/GMRES method with the length of the horizon $T(t)$ increased smoothly toward T_f.

Entire algorithm

Algorithm 2 shows the entire algorithm of the moving switching sequence approach. Note that, in Algorithm 2, we treat q_1 as the first element of σ and q_m as the last element of σ. This algorithm basically updates $U(t)$ using the C/GMRES method introduced in Section 4.1. If we predict that $t_1(t + \Delta t)$ satisfies $t_1(t + \Delta t) < t + \Delta t$, where q_1 is the first element of the switching sequence σ, we regard that a switch occurs between the current sampling time and the next sampling time and we remove variables related to the switch from $U(t)$ and remove q_1 from σ. While computing $x_0^*(t), \dots, x_N^*(t)$ in solving the TPBVP, we monitor whether a new switch occurs or not at $t' \in (t_{m-1}, t + T]$, where q_m is the last element of σ. If we detect a switch from q_m to q_{m+1} for some $q_{m+1} \in Q$ at $t' \in (t_{m-1}, t + T]$, we obtain the additional variables related to the switch numerically and reconstruct $U(t)$ by Algorithm 1 and append q_{m+1} to σ. If we detect the violation of Assumption 6, we shrink the horizon and modify $U(t)$ and σ.

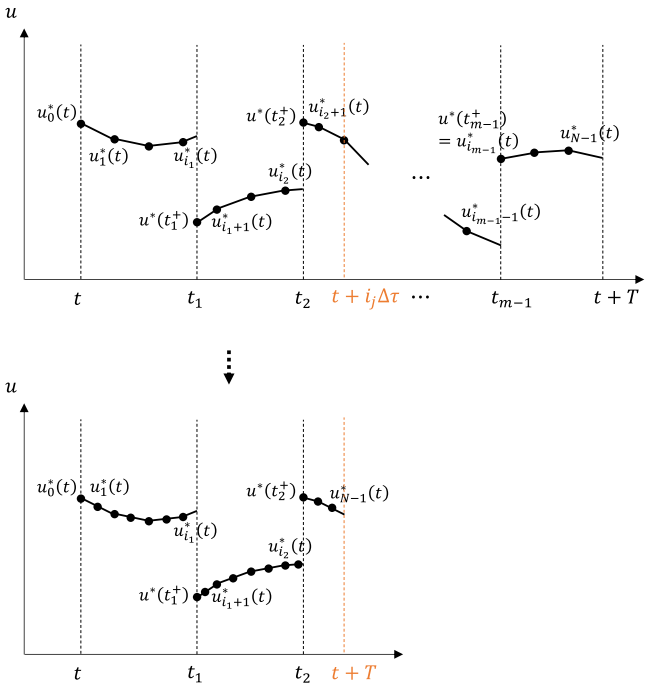

FIGURE 5 Linear interpolation of $u_0^*(t), \dots, u_{N-1}^*(t)$ when shrinking the horizon [Colour figure can be viewed at wileyonlinelibrary.com]

Algorithm 2 The moving switching sequence approach

1: Set $t = 0$ and measure $x(0)$ and the active subsystem at time 0, and compute $U(0)$ on the basis of the active subsystem at $t = 0$ and $x(0)$ by Newton's method for (66) and input $u(t) = u_0^*(t)$ to the actual system.
2: Set $t := t + \Delta t$ and observe $x(t)$.
3: Predict state $x_0^*(t), \dots, x_N^*(t)$ on the basis of $x(t)$, $U(t)$, and σ.
4: **if** the violation of Assumption 6 is detected **then**
5: Shrink the horizon and modify $U(t)$ and σ
6: **end if**
7: **if** a switch from subsystem q_m to q_{m+1} not observed at the previous sampling time is detected at $t' \in (t_m, t + T]$ **then**
8: Reconstruct $U(t)$ and σ by Algorithm 1
9: **end if**
10: Compute $\lambda_N^*(t), \dots, \lambda_1^*(t)$ on the basis of $U(t)$ and $x_0^*(t), \dots, x_N^*(t)$.
11: Compute $x_0^*(t + h), \dots, x_0^*(t + h)$ and $\lambda_N^*(t + h), \dots, \lambda_1^*(t + h)$.
12: Solve (40) by using the FD-GMRES method and obtain $\dot{U}(t)$.
13: Compute $U(t + \Delta t)$ by integrating $\dot{U}(t)$ as stated in Section 4.1.
14: **if** $t_1(t + \Delta t) < t + \Delta t$ **then**
15: Remove $U_{q_1,q_2}(t)$ from $U(t + \Delta t)$ and remove q_1 from σ
16: **end if**
17: Input $u(t) = u_0^*(t)$ to the actual system and go back to step 2.

5 | NUMERICAL SIMULATIONS OF COMPASS-LIKE WALKING

This section describes an application of the proposed method to the simplest biped walking robot model, called a compass-like model.[2] Compass-like walking involves state-dependent switches and state jumps, and we successfully control it by using the moving switching sequence approach.

5.1 | Compass-like walking model

Figure 6 shows the model of the compass-like robot, and Table 1 lists its physical parameters. The compass-like robot is composed of two legs and one joint, and we assume that the physical properties of both legs are the same. Let θ_1 be

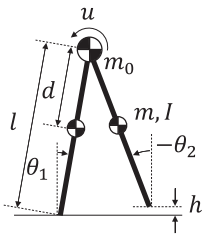

FIGURE 6 Compass-like walking model

TABLE 1 Physical parameters of compass-like model

Mass of the joint	$m_0 = 1.0$ [kg]
Mass of a leg	$m = 1.0$ [kg]
Length of a leg	$l = 1.0$ [m]
Distance between the joint and the center of mass of a leg	$d = 0.5$ [m]
Inertia of a leg	$I = 0.08333$ [kg·m^2]

the angle of leg 1 in the vertical direction, θ_2 be that of leg 2, $q = [\theta_1\, \theta_2]^{\mathrm{T}}$ be the angle vector, and $x = [q^{\mathrm{T}}\, \dot{q}^{\mathrm{T}}]^{\mathrm{T}}$ be the state vector. The dynamics of compass-like walking contains three phases: (1) leg 1 is standing on the ground and leg 2 is swinging, (2) leg 2 is standing on the ground and leg 1 is swinging, and (3) legs 1 and 2 are standing on the ground. To simplify the problem, we make the following assumption about the impact between the swinging leg and the ground.[3]

Assumption 8. The impact between the swinging leg and the ground results in no rebound and no slipping, and the supporting leg lifts from the ground at the time of the impact.

Under this assumption, the model of the compass-like walking robot is composed of the dynamics when one leg is standing on the ground and the other is swinging and state jumps by the collisions between the swinging leg and the ground. The state equation when leg 1 is standing and leg 2 is swinging is derived by Lagrange's method as

$$\dot{x} = f_1(x, u) := \begin{bmatrix} \dot{q} \\ M_1^{-1}(q)\left(B_1 u - H_1(q, \dot{q})\right) \end{bmatrix}, \tag{69}$$

where $u \in \mathbb{R}$ is a control input torque, $M_1 \in \mathbb{R}^{2\times 2}$ is the inertia matrix, $H_1 \in \mathbb{R}^2$ consists of terms related to Coriolis force and terms about gravity force, and $B_1 = [1\ -1]^{\mathrm{T}}$. Additionally, the state equation when leg 2 is standing and leg 1 is swinging is derived as

$$\dot{x} = f_2(x, u) := \begin{bmatrix} \dot{q} \\ M_2^{-1}(q)\left(B_2 u - H_2(q, \dot{q})\right) \end{bmatrix}, \tag{70}$$

where $M_2 \in \mathbb{R}^{2\times 2}$ is the inertia matrix, $H_2 \in \mathbb{R}^2$ consists of terms related to Coriolis force and terms about gravity force, and $B_2 = [-1\ 1]^{\mathrm{T}}$.

The equations of state jump are derived from the conservation law of angular momentum.[2] When leg 2 collides with the ground, the conservation law of angular momentum is written as

$$Q_{1,2}^{+}\left(q\left(t_c^{+}\right)\right)\dot{q}\left(t_c^{+}\right) = Q_{1,2}^{-}\left(q\left(t_c^{-}\right)\right)\dot{q}(t_c^{-}), \tag{71}$$

where $Q_{1,2}^{-}, Q_{1,2}^{+} \in \mathbb{R}^{2\times 2}$, and when leg 1 collides to the ground, the conservation law of angular momentum is written as

$$Q_{2,1}^{+}\left(q\left(t_c^{+}\right)\right)\dot{q}\left(t_c^{+}\right) = Q_{2,1}^{-}\left(q(t_c^{-})\right)\dot{q}(t_c^{-}), \tag{72}$$

where $Q_{2,1}^{-}, Q_{2,1}^{+} \in \mathbb{R}^{2\times 2}$. Note that the posture of the robot is not changed by the collision because the collision is instantaneous from Assumption 8. The equations of the state jump are then given as

$$x\left(t_c^{+}\right) = \gamma_{1,2}(x(t_c^{-})) := \begin{bmatrix} q(t_c^{-}) \\ \left(Q_{1,2}^{+}\left(q\left(t_c^{+}\right)\right)\right)^{-1} Q_{1,2}^{-}(q(t_c^{-}))\dot{q}(t_c^{-}) \end{bmatrix}, \tag{73}$$

and

$$x\left(t_c^+\right) = \gamma_{2,1}(x(t_c^-)) := \begin{bmatrix} q(t_c^-) \\ \left(Q_{2,1}^+\left(q\left(t_c^+\right)\right)\right)^{-1} Q_{2,1}^-(q(t_c^-))\dot{q}(t_c^-) \end{bmatrix}. \tag{74}$$

The collision occurs when the height of the tip of the swinging leg, h in Figure 6, becomes zero. That is, the switching conditions are given as

$$\psi_{1,2}(x) := l(\cos\theta_1 - \cos\theta_2) = 0, \tag{75}$$

and

$$\psi_{2,1}(x) := l(\cos\theta_2 - \cos\theta_1) = 0. \tag{76}$$

We introduce an assumption to abstract compass-like walking motion easily as a switched system.[3]

Assumption 9. The swinging leg does not contact with the ground when the swinging leg is put forward from behind the supporting leg.

Under Assumption 9, we ignore the collision between leg 2 and the ground when $\theta_1 - \theta_2 \leq 0$ and the collision between leg 1 and the ground when $\theta_2 - \theta_1 \leq 0$. Figure 7 summarizes switches and state jumps in compass-like walking.

5.2 | Controller design

To achieve a steady gait, we design the cost function with two aims: moving forward steadily and achieving a steady gait. To achieve the former aim, we make the angular velocity of the supporting leg close to the reference constant v_{ref}. To achieve the latter aim, we add $a_2(\theta_1 + \theta_2)^2$ to the cost function, thereby preventing the tip of the swinging leg from being too high. In addition to these two aims, we add a penalty to the control input u. Consequently, we design the cost functions as

$$L_1(x,u) = \frac{1}{2}a_1(\dot{\theta}_1 - v_{\text{ref}})^2 + \frac{1}{2}a_2(\theta_1+\theta_2)^2 + \frac{1}{2}ru^2, \tag{77}$$

$$L_2(x,u) = \frac{1}{2}a_1(\dot{\theta}_2 - v_{\text{ref}})^2 + \frac{1}{2}a_2(\theta_1+\theta_2)^2 + \frac{1}{2}ru^2, \tag{78}$$

and $\varphi_1(x) = \varphi_2(x) = 0$. The parameters of NMPC are listed in Table 2.

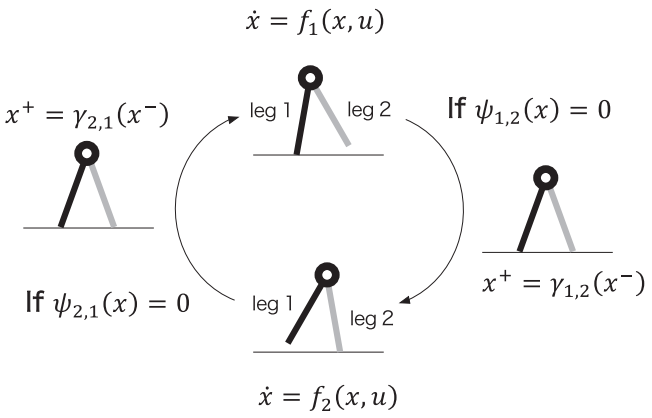

FIGURE 7 State-dependent switches and state jumps in compass-like walking

TABLE 2 Parameters of nonlinear model predictive control

A weight parameter in the cost function	$a_1 = 1.0$
A weight parameter in the cost function	$a_2 = 1.0$
A weight parameter in the cost function	$r = 0.5$
The reference velocity in the cost function	$v_{\text{ref}} = 0.5$ [rad/s]
A parameter on the length of the horizon	$T_f = 0.8$ [s]
A parameter on the length of the horizon	$\alpha = 1.0$
Number of divisions of the horizon	$N = 80$
Time increment in forward-difference generalized minimum residual (GMRES) method	$h = 1.0 \times 10^{-9}$
The stabilization parameter in (40)	$\zeta = 1000$
Maximum number of iterations in the GMRES	$k_{\max} = 5$
Convergent radius of Newton's method in the reconstruction of the solution	$r_{\text{init}} = 1.0 \times 10^{-7}$
Maximum number of iterations in Newton's method in the reconstruction of the solution	$i_{\max} = 5$

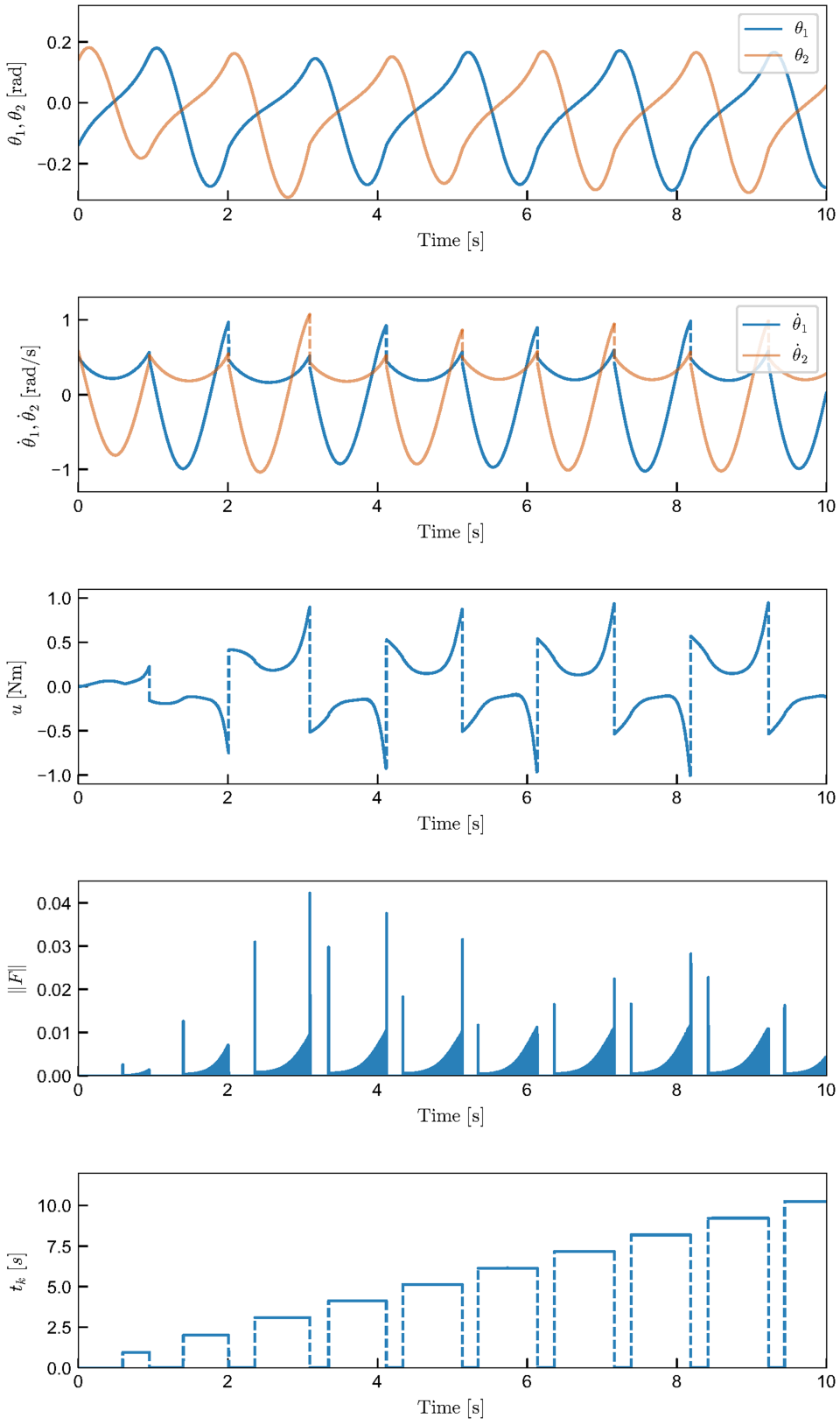

FIGURE 8 Simulation result of the nominal case [Colour figure can be viewed at wileyonlinelibrary.com]

5.3 | Simulation conditions

To evaluate the robustness of the proposed method, the numerical simulations are performed in two cases: the nominal case without disturbances and the case with impulsive disturbances. In the simulation with disturbances, we input instantaneous changes on the angular velocity as $\dot{\theta}_1^+ = \dot{\theta}_1^- + \Delta_1$, $\dot{\theta}_2^+ = \dot{\theta}_2^- + \Delta_2$ ($\Delta_1, \Delta_2 \in \mathbb{R}$), corresponding to impulsive

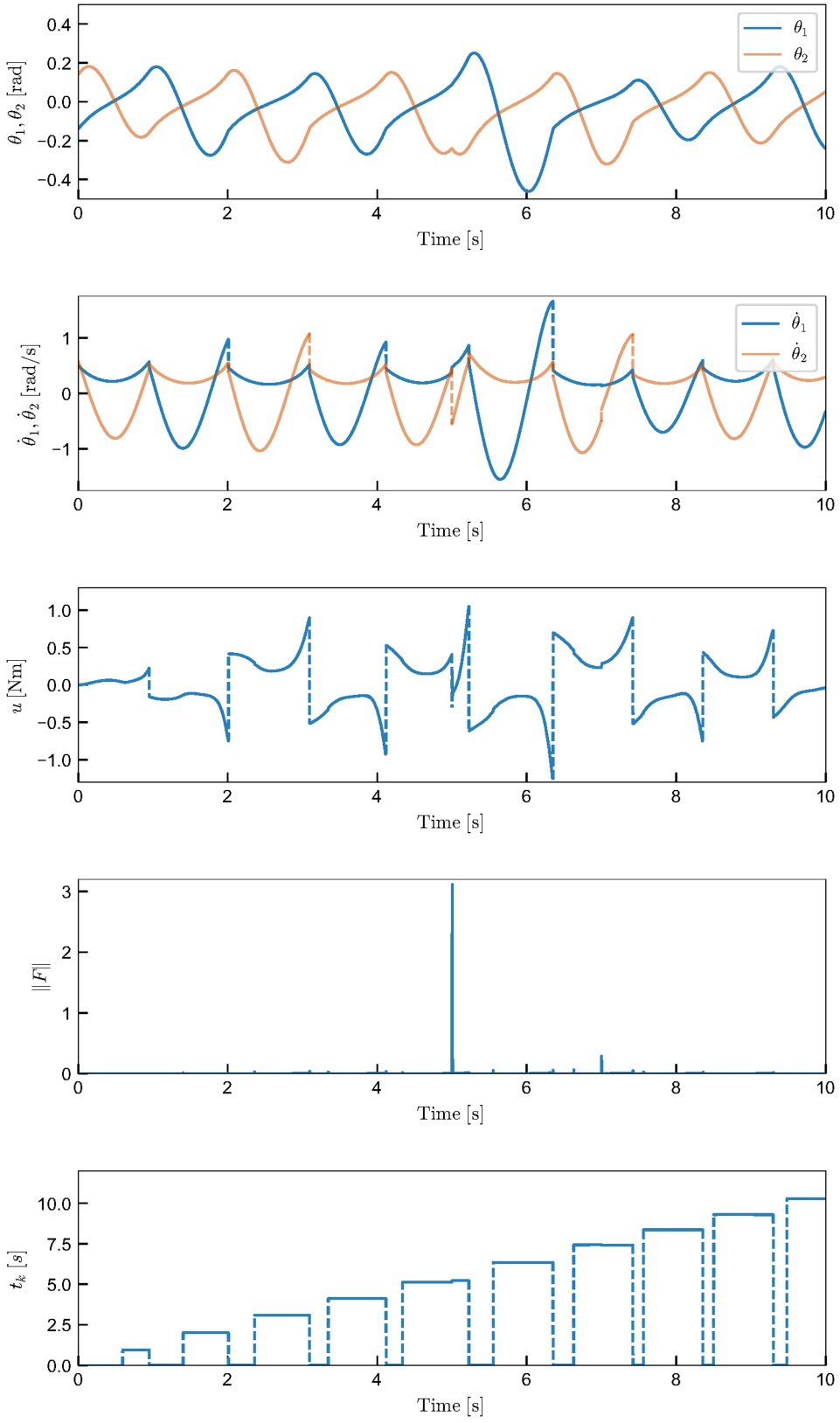

FIGURE 9 Simulation result of the case with disturbances [Colour figure can be viewed at wileyonlinelibrary.com]

disturbances. We input instantaneous changes as $\Delta_1 = 0.1$, $\Delta_2 = -1$ at $t = 5.0$ [s] and $\Delta_1 = -0.02$, $\Delta_2 = 0.2$ at $t = 7$ [s]. The former and latter represent the cases the robot is pushed from the back and the front, respectively. Both cases are simulated on the sampling period of 1 [ms], and the central processing unit (CPU) is Intel Core i5 1.8 GHz.

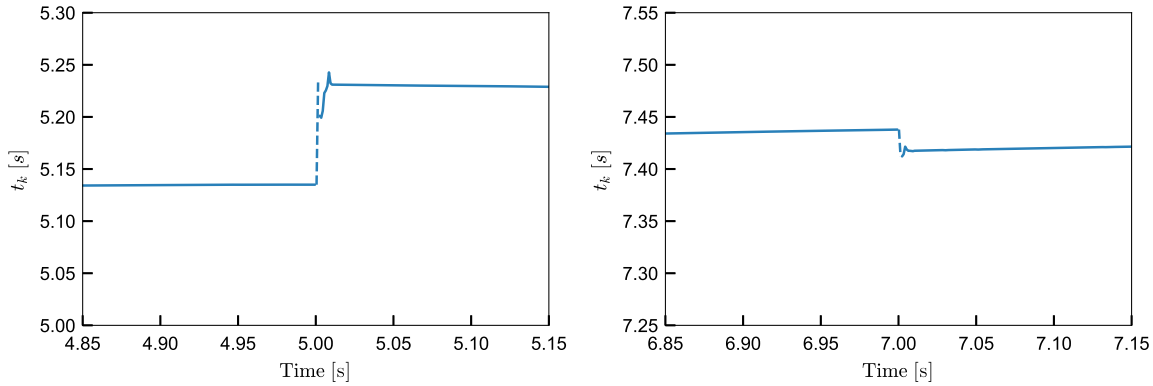

FIGURE 10 Predicted switching instants around the disturbances [Colour figure can be viewed at wileyonlinelibrary.com]

5.4 | Simulation results

Figures 8, 9, and 10 show simulation results for the nominal case, those for the case with disturbances, and the enlarged time history of the predicted switching instant around the disturbances, respectively. Note that dashed lines in these Figures represent discontinuous change of variables. In Figures 8 and 9, switches and state jumps occur at points where $\dot{\theta}_1$ and $\dot{\theta}_2$ are discontinuous except for discontinuous points due to the disturbances, and the compass-like robot walked successfully in both cases. $\|F\|$ in Figures 8 and 9 denotes the norm of the residual in (37), and t_k in Figures 8, 9, and 10 are switching instants on the predicted horizon. Note that, when there are no switches on the predictive horizon, t_k is plotted as $t_k = 0$.

We can see that the predicted switching instants are mostly constant in the nominal case from Figure 8 and fluctuate in the case with disturbances from Figures 9 and 10. From Figure 9, instantaneous increases of $\|F\|$ at the moments of the impulsive disturbances attenuate immediately, which suggests that the proposed method optimizes the switching instants and control input simultaneously. Note that, even if in the nominal case, the error increases at the sampling time when an additional switch is detected and when a switch is removed from σ, ie, when an actual switch occurs. This suggests that errors increase when the switching sequence changes from the previous sampling time. In addition to the points where the error peaks due to such reasons, we can see errors increase when there is a switch on the horizon. This occurs when $i_k(t)$ differs from $i_k(t - \Delta t)$. However, $\|F\|$ is sufficiently small on the whole, and we can see that the proposed method can compute the optimal solution. We also found that each update takes around 0.14 [ms] and that we can implement control update within the sampling period of 1 [ms].

6 | CONCLUSIONS

In this paper, we proposed a moving switching sequence approach for real-time computation of NMPC for switched systems with state-dependent switches. In the approach, the switching sequence on the horizon moves to the present time at each time as well as the optimal state trajectory and optimal control input on the horizon. We assume that the switching sequence is basically invariant on the horizon and update the switching sequence at each time by checking whether an additional switch occurs or not and whether the actual switch will occur or not. We propose an algorithm composed of the update of the solution by solving the TPBVP and reconstruction of the solution taking into account an additional switch. We utilize the C/GMRES method for the former problem and solve the reduced optimal control problem for the latter. We performed numerical simulations of compass-like walking for a nominal case without disturbances and a case with impulsive disturbances and found that the proposed method successfully computes the optimal solution in a short time even if there are disturbances.

For future work, we will analyze the necessary or sufficient conditions of the assumption about the invariance of the switching sequence. We will also seek methods to design the cost functions and constraints so that the assumption holds.

ACKNOWLEDGEMENT

This work was partly supported through the JSPS Kakenhi grant JP15H02257.

ORCID

Sotaro Katayama https://orcid.org/0000-0002-0051-2378
Toshiyuki Ohtsuka https://orcid.org/0000-0003-3554-8933

REFERENCES

1. Zhu F, Antsaklis PJ. Optimal control of hybrid switched systems: a brief survey. *Discrete Event Dyn Syst*. 2015;25(3):345-364.
2. Goswami A, Thuilot B, Espiau B. Compass-Like Biped Robot Part I: Stability and Bifurcation of Passive Gaits. Research Report. INRIA; 1996.
3. Grizzle JW, Abba G, Plestan F. Asymptotically stable walking for biped robots: analysis via systems with impulse effects. *IEEE Trans Autom Control*. 2001;46(1):51-64.
4. Yashima M, Yamaguchi H. Dynamic motion planning whole arm grasp systems based on switching contact modes. In: Proceedings of the 2002 IEEE International Conference on Robotics and Automation; 2002; Washington, DC.
5. Lengagne S, Vaillant J, Yoshida E, Kheddar A. Generation of whole-body optimal dynamic multi-contact motions. *Int J Robotics Res*. 2013;32(9-10):1104-1119.
6. Ferrara A, Paderno J. Application of switching control for automatic pre-crash collision avoidance in cars. *Nonlinear Dynamics*. 2006;46(3):307-321.
7. Gans NR, Hutchinson SA. Stable visual servoing through hybrid switched-system control. *IEEE Trans Robotics*. 2007;23(3):530-540.
8. Jin J, Kim Y, Wee S, Lee D, Gans N. A stable switched-system approach to collision-free wheeled mobile robot navigation. *J Intell Robotic Syst*. 2017;86(3):599-616.
9. Ngo V, Hofman T, Steinbuch M, Serrarens A. Optimal control of the gearshift command for hybrid electric vehicles. *IEEE Trans Veh Technol*. 2012;61(8):3531-3543.
10. Geyer T, Papafotiou G, Morari M. On the optimal control of switch-mode DC-DC converters. In: *Hybrid Systems: Computation and Control*. Berlin, Germany: Springer; 2004:342-356.
11. Kouro S, Cortes P, Vargas R, Ammann U, Rodriguez J. Model predictive control–a simple and powerful method to control power converters. *IEEE Trans Ind Electron*. 2009;56(6):1826-1838.
12. Yazdi MB, Jahed-Motlagh MR. Stabilization of a CSTR with two arbitrarily switching modes using modal state feedback linearization. *Chem Eng J*. 2009;155(3):838-843.
13. Heidarinejad M, Liu J, Christofides PD. Economic model predictive control of switched nonlinear systems. *Syst Control Lett*. 2013;62(1):77-84.
14. Rawlings JB, Meadows ES, Muske KR. Nonlinear model predictive control: a tutorial and survey. In: Proceedings of the IFAC Symposium on Advanced Control of Chemical Processes; 1994; Kyoto, Japan.
15. Mayne D. Nonlinear model predictive control: challenges and opportunities. In: Allgöwer F, Zheng A, eds. *Nonlinear Model Predictive Control*. Basel, Switzerland: Birkhäuser; 2000:23-44. *Progress in Systems and Control Theory*; vol. 26.
16. Grüne L, Pannek J. *Nonlinear Model Predictive Control: Theory and Algorithms*. London, UK: Springer; 2011.
17. Shaikh MS, Caines PE. On the hybrid optimal control problem: theory and algorithms. *IEEE Trans Autom Control*. 2007;52(9):1587-1603.
18. Long CE, Polisetty PK, Gatzke EP. Deterministic global optimization for nonlinear model predictive control of hybrid dynamic systems. *Int J Robust Nonlinear Control*. 2006;17(13):1232-1250.
19. Wang X, Mahalec V, Qian F. Globally optimal nonlinear model predictive control based on multi-parametric disaggregation. *J Process Control*. 2017;52:1-13.
20. Gokbayrak K, Cassandras CG. Hybrid controllers for hierarchically decomposed systems. In: *Hybrid Systems: Computation and Control*. Berlin, Germany: Springer; 2000:117-129.
21. Xu X, Antsaklis PJ. Optimal control of switched systems via non-linear optimization based on direct differentiations of value functions. *Int J Control*. 2002;75(16-17):1406-1426.
22. Xu X, Antsaklis PJ. Optimal control of switched systems based on parameterization of the switching instants. *IEEE Trans Autom Control*. 2004;49(1):2-16.
23. Fotiou IA, Rostalski P, Parrilo PA, Morari M. Parametric optimization and optimal control using algebraic geometry methods. *Int J Control*. 2006;79(11):1340-1358.
24. Fotiou IA, Beccuti AG, Papafotiou G, Morari M. Optimal control of piece-wise polynomial hybrid systems using cylindrical algebraic decomposition. In: *Hybrid Systems: Computation and Control*. Berlin, Germany: Springer; 2006:227-241.
25. Wei S, Uthaichana K, Žefran M, DeCarlo RA, Bengea S. Applications of numerical optimal control to nonlinear hybrid systems. *Nonlinear Anal Hybrid Syst*. 2007;1(2):264-279.
26. Meyer RT, Žefran M, DeCarlo RA. A comparison of the embedding method with multiparametric programming, mixed-integer programming, gradient-descent, and hybrid minimum principle-based methods. *IEEE Trans Control Syst Technol*. 2014;22(5):1784-1800.
27. Grammel G. Maximum principle for a hybrid system via singular perturbations. *SIAM J Control Optim*. 1999;37(4):1162-1175.
28. Sussmann HJ. A maximum principle for hybrid optimal control problems. In: Proceedings of the 38th IEEE Conference on Decision and Control; 1999; Phoenix, AZ.
29. Riedinger P, Iung C, Kratz F. An optimal control approach for hybrid systems. *Eur J Control*. 2003;9(5):449-458.

30. Kaya CY, Noakes JL. Computational method for time-optimal switching control. *J Optim Theory Appl*. 2003;117(1):69-92.
31. Hwang I, Li J, Du D. A numerical algorithm for optimal control of a class of hybrid systems: differential transformation based approach. *Int J Control*. 2008;81(2):277-293.
32. Taringoo F, Caines PE. Gradient geodesic and newton geodesic HMP algorithms for the optimization of hybrid systems. *Annu Rev Control*. 2011;35(2):187-198.
33. Taringoo F, Caines PE. On the optimal control of impulsive hybrid systems on Riemannian manifolds. *SIAM J Control Optim*. 2013;51(4):3127-3153.
34. Pakniyat A, Caines PE. On the relation between the minimum principle and dynamic programming for classical and hybrid control systems. *IEEE Trans Autom Control*. 2017;62(9):4347-4362.
35. Pakniyat A, Caines PE. Hybrid optimal control of an electric vehicle with a dual-planetary transmission. *Nonlinear Anal Hybrid Syst*. 2017;25:263-282.
36. Cannon M. Efficient nonlinear model predictive control algorithms. *Annu Rev Control*. 2004;28(2):229-237.
37. Ohtsuka T. A continuation/GMRES method for fast computation of nonlinear receding horizon control. *Automatica*. 2004;40(4):563-574.
38. Nikol'skii MS. Convergence of the gradient projection method in optimal control problems. *Comput Math Model*. 2007;18(2):148-156.
39. Schäfer A, Kühl P, Diehl M, Schlöder J, Bock H. Fast reduced multiple shooting methods for nonlinear model predictive control. *Chem Eng Process Process Intensif*. 2007;46(11):1200-1214.
40. Diehl M, Ferreau HJ, Haverbeke N. Efficient numerical methods for nonlinear MPC and moving horizon estimation. In: *Nonlinear Model Predictive Control*. Berlin, Germany: Springer; 2009:391-417.
41. Neunert M, de Crousaz C, Furrer F, Kamel M, Farshidian F, Siegwart R, Buchli J. Fast nonlinear model predictive control for unified trajectory optimization and tracking. In: Proceedings of the 2016 IEEE International Conference on Robotics and Automation; 2016; Stockholm, Sweden.
42. Cisneros PG, Werner H. Fast nonlinear MPC for reference tracking subject to nonlinear constraints via quasi-LPV representations. *IFAC Pap*. 2017;50(1):11601-11606. Part of special issue: 20th IFAC World Congress.
43. Yamakita M, Taura A, Onodera Y. An application of nonlinear receding horizon control to posture control with collisions. In: Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics; 2005; Monterey, CA.
44. Onodera Y, Yamakita M. An extension of nonlinear receding horizon control for switched system with state jump. In: Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2005; Edmonton, Canada.
45. Katayama S, Satoh Y, Doi M, Ohtsuka T. Nonlinear model predictive control for systems with autonomous state jumps using a penalty function method. In: Proceedings of the 11th Asian Control Conference; 2017; Gold Coast, Canada.
46. Bryson AE, Ho YC. *Applied Optimal Control*. New York, NY: Taylor and Francis; 1975.
47. Richter SL, Decarlo RA. Continuation methods: theory and applications. *IEEE Trans Syst Man Cybern*. 1983;SMC-13(4):459-464.
48. Kelley CT. *Iterative Methods for Linear and Nonlinear Equations*. Philadelphia, PA: Society for Industrial Applied Mathematics; 1995.

How to cite this article: Katayama S, Doi M, Ohtsuka T. A moving switching sequence approach for nonlinear model predictive control of switched systems with state-dependent switches and state jumps. *Int J Robust Nonlinear Control*. 2020;30:719–740. https://doi.org/10.1002/rnc.4804