IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 16, 2019, accepted December 29, 2019, date of publication January 13, 2020, date of current version January 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966240

Real-Time Model Predictive Control of Rigid Body Motion via Discretization Using the Cayley Map

YUICHI TADOKORO, (Member, IEEE), YUKI TAYA, TATSUYA IBUKI, (Member, IEEE), AND MITSUJI SAMPEI, (Member, IEEE)
Tokyo Institute of Technology, Tokyo 152-8550, Japan

Corresponding author: Yuichi Tadokoro (tadokoro@sl.sc.e.titech.ac.jp)

This work was supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant 16H04383 and Grant 18J14658.

ABSTRACT This paper presents a fast nonlinear model predictive control method for rigid body dynamical systems such as spacecraft or aerial vehicles on the special Euclidean group SE(3). The focus of this research is on the real-time execution of the optimal control on a low-cost embedded computer. The position and orientation of a rigid body are expressed as a 6-dimensional vector via the Cayley map for SE(3). Based on the representation, the motion can be exactly discretized on the algebra se(3). As a result, coarse sampling intervals can be selected to reduce the number of prediction steps. Furthermore, the recursive discretization technique is applied to effectively reduce the decision variables of the nonlinear optimization problem by eliminating states from them. Simulation results on a Raspberry Pi single-board computer are given to prove that the present model predictive control method is feasible in real time. The effectiveness of the present controller is further verified by an experiment using a fully actuated hexarotor unmanned aerial vehicle.

INDEX TERMS Optimal control, robot control, nonlinear control systems, special Euclidean group.

I. INTRODUCTION

Rigid body kinematics and dynamics are the fundamentals of the modeling, control, and measurement of aerial robots, spacecraft, robot manipulators, etc. The rigid body motion is described with a dynamical model in the special Euclidean group $SE(3)$. The dynamics must be constrained on the group, and thus, we need to consider specific control laws to achieve regulation and tracking. For example, [1] presents coordinate-free geometric PD controllers on $SE(3)$ and its subgroup $SO(3)$, the special orthogonal group, based on the metric structure of the groups. Lee *et* al. [2] study a trajectory tracking controller of a quadrotor aerial vehicle on $SE(3)$ using an intrinsic attitude tracking error defined on $SO(3)$. Such a control strategy can also be applied for optimal control problems. Liu *et* al. [3] show an analytic solution of a quadratic optimal control problem in a compact Lie group such as $SO(3)$ by using the logarithm map. The authors of this paper have presented an analytic optimal control method for $SE(3)$ considering a quadratic objective function with specific weight matrices [4]. It also shows an application of the control of a fully actuated hexarotor aerial vehicle (Fig. 1) which can be identified with a fully actuated rigid body.

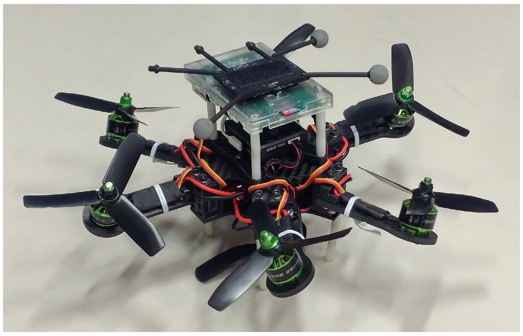


FIGURE 1. Fully actuated hexarotor aerial vehicle used in the experiment. Each rotor is tilted and fixed so that the vehicle can generate force and torque in every direction and around any axis.

In general, however, it is difficult to solve optimal control problems analytically. For example, [4] cannot consider external force such as the force of gravity, and the objective function is limited to a certain form. Hence, we need numerical methods if we consider constraints or complex objective functions. The key to numerical optimal control for rigid body motion is proper discretization of the dynamics constrained on Lie groups. The simplest possible way of discretization is a forward Euler approximation in a vector space [5]. Although this approach can be sufficient for many

The associate editor coordinating the review of this manuscript and approving it for publication was Ning Sun.

other cases, it is not recommended for rigid body systems because the Lie group structure is not preserved after the integration. In related research [6]–[8], geometric integrators are used to overcome this problem. Lee *e*t al. [6] consider structure-preserving optimal control of discrete-time rigid body systems. They propose a computational approach based on a Lie group variational integrator (LGVI) and Newton methods to solve a minimum-energy control problem. Kobilarov and Marsden [7] study a generalized method of LGVI-based discrete optimal control for both fully actuated and underactuated systems on any Lie group. In these studies, conditions regarding the variational principle and optimality are combined into a system of nonlinear equations, and the optimal control problem becomes equivalent to a nonlinear root-finding problem. However, LGVI-based optimal control methods usually require optimization over both states (or costates) and inputs. This results in a larger scale problem that can be intractable on embedded controllers in real time due to the curse of dimensionality. Moreover, these methods are only concerned with minimum-energy control problems where the objective function is limited to the total of the squared input norm.

For the purpose of developing a more versatile optimal control method, we adopt nonlinear model predictive control (NMPC) with a geometric integrator. In particular, we focus on fast computation of NMPC for embedded controllers in real time. To develop such a numerical optimal control method, we need to reduce the number of decision variables while preserving the Lie group structure. Instead of optimizing the sequence of both states and inputs, we employ the so-called recursive discretization technique [9, Chapter 10] to eliminate states from the decision variables. To apply the technique, we use a geometric integrator based on the Cayley map for $SE(3)$ [8], [10] instead of an LGVI. The geometric integrator with the Cayley map preserves the Lie group structure, and it enables us to pick a coarse sampling interval for the prediction and to reduce the number of decision variables. Next, we calculate the analytic gradient of the objective function with respect to the sequence of inputs, exploiting the sparsity of intermediate Jacobian matrices. The novel usage of the geometric integrator together with the recursive discretization technique reduces the computational effort while maintaining the precision of the predicted trajectory.

The main contribution of this paper is to develop a real-time NMPC method for wide variety of single rigid body systems in 3-dimensional space such as spacecraft and unmanned aerial vehicles [4]. The rigid body dynamics model includes an external force term such as the force of gravity or airflows. The contribution is notable because this work includes a real-time on-board NMPC experiment for an aerial vehicle system that has fast dynamics and inherent instability. In fact, many NMPC schemes for mobile robot systems presented in [7], [11]–[14], for example, show that various control objectives are achieved but have no experimental validations. Moreover, experiments of NMPC usually employ high-performance processors, e.g., Intel Core i7 CPUs, for calculation on the ground [15]–[17]. To the author's knowledge, there are only a few exceptions such as a study on NMPC for aerial manipulation using an aerial vehicle [18] and their later work on obstacle avoidance control [19]. Although they perform experiments with NMPC computation on an embedded computer, the optimal control law is used to generate the reference trajectory, and it is not used for real-time control. In this paper, on the other hand, we tackle real-time control of such an unstable system with an NMPC running on an embedded controller.

The rest of the paper is organized as follows. In Section II, we first introduce a continuous-time model of a rigid body dynamical system. Then, structure-preserving geometric integrators based on the exponential map and the Cayley map for $SE(3)$ are explained. Based on the Cayley map, we present a discrete-time model of the rigid body dynamics in Section III. In Section IV, the discrete-time model is used to formulate an NMPC problem. A recursive discretization technique is applied to reduce the number of decision variables, and the gradient of the objective function is analytically obtained to further speed up the optimization. Simulations are performed on a Raspberry Pi single-board computer in Section V, and the results show that the present fast NMPC is feasible in real time on the low-cost embedded system. Finally, in Section VI, the effectiveness of the present NMPC method is further verified through a flight control experiment of a fully actuated hexarotor aerial vehicle. Section VII concludes this paper.

II. PRELIMINARIES

This section describes mathematical notion of rigid body dynamics as the target system of this research. In this paper, we suppose that the body is not deformable, and the rigid body has actuators such as thrusters or propellers so that the vehicle can independently control 6 degrees of freedom of the translational and rotational motions in the 3-dimensional space.

A. MATHEMATICAL REPRESENTATION OF RIGID BODY MOTION

In this paper, O_n and $O_{n\times m}$, respectively, denote the $n \times n$ and $n \times m$ zero matrices, and I_n denotes the $n \times n$ identity matrix. $R_x(\theta)$, $R_y(\theta)$, and $R_z(\theta)$ denote rotation matrices by the angle θ around x, y, and z axes, respectively. For the detailed explanation on the rigid body dynamics expressed on Lie groups, the reader is referred to [20].

The special orthogonal group $SO(3)$ is a set of 3-dimensional rotation matrices

$$SO(3) = \left\{ R \in \mathbb{R}^{3\times 3} \mid R^\top R = RR^\top = I_3 \wedge \det R = 1 \right\}.$$

The special Euclidean group $SE(3)$ is a semidirect product of $SO(3)$ and $\mathbb{R}^3$, and an element of $SE(3)$ represents a pair of a rotation matrix and a position vector in 3-dimensional space. From the definition, the pose (position and orientation) of a

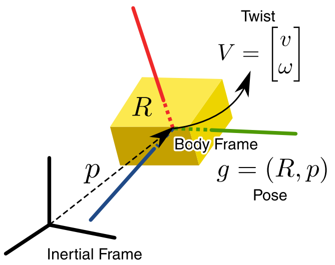


FIGURE 2. **Pose and twist of a rigid body. The pose $g \in SE(3)$ is defined as a pair of the rotation matrix $R \in SO(3)$ representing the attitude of the body and the position vector $p \in \mathbb{R}^3$. The pose is driven by the twist $V \in \mathbb{R}^6$ representing simultaneous linear and angular motion of the body.**

rigid body shown in Fig. 2 is expressed with an element of $SE(3)$ as

$$g = (R, p) = \begin{bmatrix} R & p \\ O_{1\times 3} & 1 \end{bmatrix} \in SE(3),$$

where $R \in SO(3)$ and $p \in \mathbb{R}^3$ denote a rotation matrix and a position vector, respectively. The matrix is referred to as the homogeneous transformation matrix, and it represents both the pose and coordinate transformation. The Lie algebras $\mathfrak{so}(3)$ and $\mathfrak{se}(3)$ related to the infinitesimal changes on groups $SO(3)$ and $SE(3)$, respectively, consist of the matrices

$$\widehat{\omega} = (\omega)^\wedge = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathfrak{so}(3)$$

and

$$\widehat{V} = (V)^\wedge = \begin{bmatrix} \widehat{\omega} & v \\ O_{1\times 3} & 0 \end{bmatrix} \in \mathfrak{se}(3),$$

where $v \in \mathbb{R}^3$ corresponds to the linear velocity, $\omega = [\omega_1\ \omega_2\ \omega_3]^\top \in \mathbb{R}^3$ is the angular velocity, and $V = \begin{bmatrix} v^\top & \omega^\top \end{bmatrix}^\top \in \mathbb{R}^6$ denotes the twist (linear and angular velocity) of the rigid body. Here, the "wedge" operator $(\cdot)^\wedge$ is used to identify the elements of the Lie algebras with real vectors. We also define the "vee" operator $(\cdot)^\vee$ as the inverse of the wedge operator.

Now, we consider the dynamics of a rigid body with the mass $m \in \mathbb{R}$ and the inertia tensor $\mathcal{I} \in \mathbb{R}^{3\times 3}$. Let $g \in SE(3)$ and $V = \begin{bmatrix} v^\top & \omega^\top \end{bmatrix}^\top \in \mathbb{R}^6$ be the pose and the body twist of the rigid body, respectively. Assume that the wrench (force and torque) $F \in \mathbb{R}^6$ and the force of gravity expressed in the body frame

$$F_{\text{grav}} = \begin{bmatrix} R^\top \begin{bmatrix} 0 & 0 & -mG \end{bmatrix}^\top \\ O_{3\times 1} \end{bmatrix}$$

are acting on the center of mass. Here, $G \in \mathbb{R}$ denotes the gravitational acceleration constant. Then, the rigid body dynamics evolve according to a kinematic equation on $SE(3)$ and the Newton-Euler equation of motion [20].

$$\begin{gathered} \dot{g} = g\widehat{V} \\ \begin{bmatrix} mI_3 & O_3 \\ O_3 & \mathcal{I} \end{bmatrix} \dot{V} + \begin{bmatrix} m\widehat{\omega}v \\ \widehat{\omega}(\mathcal{I}\omega) \end{bmatrix} = F - F_{\text{grav}} \end{gathered} \tag{1}$$

We transform the equations of motion by canceling the second term of (1), namely, the Coriolis force and gyroscopic torque, via feedback linearization. Finally, the dynamics are expressed as the double-integrator system on $SE(3)$ subject to gravity

$$\dot{g} = g\widehat{V}, \quad \dot{V} = u - u_{\text{grav}}, \tag{2}$$

where u is the acceleration input of the body applied to the center of mass, and $u_{\text{grav}} = \frac{1}{m}F_{\text{grav}}$.

Remark 1: We explicitly consider the force of gravity instead of simply canceling it out by using feedback linearization. This is because we would like to penalize the gravity compensation input, which consumes a significant amount of energy in flight control of aerial vehicles.

B. EXPONENTIAL MAP FOR SE(3)

Now, we discretize the continuous-time dynamical model (2) to apply a numerical optimal control scheme. The simplest method to integrate the equation (2) is to perform finite-order approximations such as forward Euler and Runge-Kutta methods in a vector space [5]. However, its time evolution of the pose cannot be constrained on $SE(3)$ when the sampling period or the twist is large. One simple approach to resolve this problem is to use the exponential map for $SE(3)$ to integrate the piecewise constant twist of a rigid body [20]. Let $V_k = \begin{bmatrix} v_k^\top & \omega_k^\top \end{bmatrix}^\top$ be the twist at the kth step that is constant for the sampling period $T > 0$. The pose g_{k+1} is then computed as

$$g_{k+1} = g_k \exp(\widehat{V}_k T), \tag{3}$$

where $\exp : \mathfrak{se}(3) \to SE(3)$, $\mathfrak{so}(3) \to SO(3)$ is given as follows:

$$e^{\widehat{V}_k T} = \begin{bmatrix} e^{\widehat{\omega}_k T} & \frac{(I_3 - e^{\widehat{\omega}_k T})(\widehat{\omega}_k v_k) + \omega_k \omega_k^\top v_k T}{\|\omega_k\|^2} \\ O_{1\times 3} & 1 \end{bmatrix}, \tag{4}$$

$$e^{\widehat{\omega}_k T} = I_3 + \frac{\sin(\|\omega_k\|T)}{\|\omega_k\|}\widehat{\omega}_k + \frac{1 - \cos(\|\omega_k\|T)}{\|\omega_k\|^2}\widehat{\omega}_k^2 \tag{5}$$

if $\|\omega_k\| \neq 0$, and otherwise,

$$e^{\widehat{V}_k T} = \begin{bmatrix} I_3 & v_k T \\ O_{1\times 3} & 1 \end{bmatrix}, \quad e^{\widehat{\omega}_k T} = I_3.$$

In other words, the exponential map generates the pose difference $g_{k+1}g_k^{-1}$ in $SE(3)$ from the constant velocity v_k and angular velocity ω_k. This geometric integration method is referred to as the Lie-Euler method [5]. (3) provides exact discrete-time evolution of the rigid body kinematics and mathematically constrains g on $SE(3)$, as shown in Fig. 3. The exponential map for $SE(3)$ described in (4) can be regarded as the extended version of Rodrigues' formula of rotation that corresponds to (5).

We now vectorize the kinematic equation (3) because the vector form is more convenient in both theory and computation than the matrix representation. Before showing our main results, we try to use the exponential map and its inverse, logarithm map [21], to obtain a vector form of (3). The

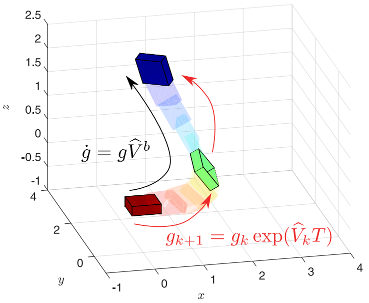


FIGURE 3. Exact discretization of rigid body motion. By using the exponential map, the continuous change in the pose can be exactly written by a recurrence formula if the twist is constant for the time period.

logarithm map log : $SE(3) \to \mathfrak{se}(3)$ is used to calculate the twist $\widehat{\psi} \in \mathfrak{se}(3)$ that is required to move the pose from the identity, i.e., $g = \exp(\widehat{\psi})$. By applying the "vee" operator to the twist, we obtain

$$\psi = \log(g)^{\vee} \in \mathbb{R}^6. \tag{6}$$

The vector ψ can be considered as a vector form of the pose g. Applying (6) to (3) yields

$$\psi_{k+1} = \log(\exp(\widehat{\psi}_k)\exp(\widehat{V}_k T))^{\vee}. \tag{7}$$

The new state ψ_{k+1} can be expressed as a function of ψ_k and V_k by using the Baker-Campbell-Hausdorff formula [22]

$$\begin{aligned}\widehat{\psi}_{k+1} &= \widehat{\psi}_k + \widehat{V}_k + \frac{1}{2}[\widehat{\psi}_k, \widehat{V}_k] \\ &\quad + \frac{1}{12}[\widehat{\psi}_k, [\widehat{\psi}_k, \widehat{V}_k]] + \frac{1}{12}[\widehat{V}_k, [\widehat{\psi}_k, \widehat{V}_k]] + \cdots,\end{aligned}$$

where $[\cdot, \cdot]$ denotes the Lie bracket. It is reported that one can obtain the closed-form equation by using dual vectors [23]. However, calculation of log needs a transcendental function $\cos^{-1}$ to extract the angle from the rotation matrix, as mentioned in [10] and [20, Appendix A]. Moreover, $\log(g)$ is not unique for an element of $SE(3)$ because the rotation angle may take multiple values $\theta + 2\pi n$, where θ is a representative value of $\log(g)$ and $n \in \mathbb{Z}$, for a rotation matrix. These characteristics of the logarithm map make it difficult to derive a simple discretized model to be used in the prediction part of NMPC.

III. EXACT DISCRETIZATION BASED ON THE CAYLEY MAP

In this paper, we discretize the rigid body dynamics based on the Cayley map for $SE(3)$ to overcome the problems of the logarithm map, which we have mentioned in the last section. There exist several definitions of the Cayley map. We refer to [10] for the definition.

Let $\widehat{\psi}^{\copyright} \in \mathfrak{se}(3)$ and $g \in SE(3)$. The Cayley map for $SE(3)$ is a mapping from the Lie algebra $\mathfrak{se}(3)$ to the group $SE(3)$ defined as follows:

$$\begin{aligned}\mathrm{Cay}(\widehat{\psi}^{\copyright}) &= (I_4 - \widehat{\psi}^{\copyright})^{-1}(I_4 + \widehat{\psi}^{\copyright}) = g, \\ \mathrm{Cay}^{-1}(g) &= (g + I_4)^{-1}(g - I_4) = \widehat{\psi}^{\copyright}.\end{aligned} \tag{8}$$

$$\begin{tikzcd}
\widehat{\psi} \in \mathfrak{se}(3) \arrow[rd, shift left, "\exp"] & \\
& g \in SE(3) \arrow[lu, shift left, "\log"] \arrow[ld, shift right, "\mathrm{Cay}^{-1}"'] \\
\widehat{\psi}^{\copyright} \in \mathfrak{se}(3) \arrow[ru, shift right, "\mathrm{Cay}"'] &
\end{tikzcd}$$

FIGURE 4. Relation among ψ, $\psi^{\copyright}$, and g. An element of $SE(3)$ can be expressed in two different ways in $\mathfrak{se}(3)$ via the exponential map and the Cayley map.

We call $\psi^{\copyright} \in \mathbb{R}^6$ the *Cayley parameter* of $g \in SE(3)$ in this paper. The symbol $(\cdot)^{\copyright}$ denotes a vector regarding the Cayley map and its derivative. Fig. 4 shows the relation among these symbols that we use throughout this paper.

A. TWIST AND ACCELERATION TRANSFORMATION

In contrast to the exponential map, the Cayley map does not represent the mapping from the physically meaningful twist to the pose. It is thus important to obtain transformations between the actual twist ψ and the Cayley parameter $\psi^{\copyright}$. Here, we define $\psi \in \mathbb{R}^6$ such that $g = \exp(\widehat{\psi})$, $\psi = \left[\xi^\top \; \eta^\top\right]^\top$ and the corresponding Cayley parameter as $\psi^{\copyright} = \left[\xi^{\copyright\top} \; \eta^{\copyright\top}\right]^\top$. We separately consider the translational and rotational parts denoted by ξ and η, respectively. Direct calculation of $\psi^{\copyright} = \mathrm{Cay}^{-1}(\exp(\widehat{\psi}))^{\vee}$ yields

$$\xi^{\copyright} = \underbrace{\left(\frac{\eta\eta^\top}{2\|\eta\|^2} - \frac{\tan\frac{\|\eta\|}{2}}{\|\eta\|}\frac{\widehat{\eta}^2}{\|\eta\|^2}\right)}_{:=\mathcal{A}(\eta)\in\mathbb{R}^{3\times3}}\xi \tag{9}$$

$$\eta^{\copyright} = \frac{\tan\frac{\|\eta\|}{2}}{\|\eta\|}\eta, \tag{10}$$

if $\eta \neq O_{3\times1}$. Likewise, $\eta^{\copyright} = O_{3\times1}$ and $\xi^{\copyright} = \frac{1}{2}\xi$ hold if $\eta = O_{3\times1}$. Note that $\lim_{\|\eta\|\to0}\psi^{\copyright} = \psi^{\copyright}\big|_{\eta=O_{3\times1}}$ holds, and thus (9) and (10) are continuous at $\eta = O_{3\times1}$.

The rotational part $\eta^{\copyright}$ in (10) corresponds to the classical Rodrigues parameter, which is one of the representations of 3-dimensional rotation [24]. In fact, we have the same relation between the Rodrigues parameter $\eta^{\copyright}$ and the corresponding rotation matrix R, as in (8).

$$R = (I_3 - \widehat{\eta}^{\copyright})^{-1}(I_3 + \widehat{\eta}^{\copyright})$$

Therefore, we can interpret $\psi^{\copyright}$, the Cayley parameter for $SE(3)$, as an extended version of the Rodrigues parameter.

We also need to derive the relation between the actual linear/angular acceleration $u = [u_{\mathrm{tr}}^\top \; u_{\mathrm{rot}}^\top]^\top \in \mathbb{R}^6$ and the corresponding acceleration in the Cayley parametrization $u^{\copyright} = [u_{\mathrm{tr}}^{\copyright\top} \; u_{\mathrm{rot}}^{\copyright\top}]^\top \in \mathbb{R}^6$. We obtain the following equations by taking the time derivatives of (9) and (10):

$$\begin{aligned}u_{\mathrm{tr}}^{\copyright} &= \frac{dv^{\copyright}}{dt} \\ &= \begin{cases}\mathcal{A}(\omega)u_{\mathrm{tr}} + \mathcal{B}(v, \omega)u_{\mathrm{rot}} & \text{for } \|\omega\| \neq 0 \\ \dfrac{1}{2}u_{\mathrm{tr}} & \text{for } \|\omega\| = 0\end{cases},\end{aligned} \tag{11}$$

$$
\begin{aligned}
u_{\mathrm{rot}}^{\copyright} &= \frac{d\omega^{\copyright}}{dt} \\
&= \begin{cases} \left(\dfrac{\omega\omega^{\top}}{(1+\cos\|\omega\|)\|\omega\|^2} - \dfrac{\tan\frac{\|\omega\|}{2}}{\|\omega\|}\dfrac{\widehat{\omega}^2}{\|\omega\|^2} \right) u_{\mathrm{rot}} & \\ \qquad\qquad\qquad \text{for } \|\omega\| \neq 0 & \\ \dfrac{1}{2}\, u_{\mathrm{rot}} \qquad\qquad \text{for } \|\omega\| = 0 & \end{cases}
\end{aligned} \tag{12}
$$

Here, $\mathcal{A}$ is defined in (9), and $\mathcal{B}(v, \omega)$ is a nonlinear matrix derived by collecting the terms of u_{rot} from $\frac{d}{dt}(\mathcal{A}(\omega)v)$. Similarly, we can also obtain the inverse transformation that is used to compute the actual acceleration input u from the virtual control input $u^{\copyright}$. These transformations are not mentioned in the related research that uses the Cayley map such as [8], but they are required in applications.

Remark 2: It is easy to see from (10) *that* $\|\eta^{\copyright}\| \to \infty$ *as* $\|\eta\| \to \pi$. *This means that the Cayley parameter cannot express the rotation by* π *rad in any axes, i.e.,* $\mathrm{tr}(R) = -1$, *and the parametrization becomes singular at such orientations. A dual quaternion [25] can be used to express the pose globally, although it is not unique for a pose and raises the same issue as the logarithm map.*

Remark 3: The Cayley parameter $\psi^{\copyright} \in \mathbb{R}^6$ *can be uniquely determined for a pose* $g \in SE(3)$ *except for the singular orientations described in Remark 2. Stabilization of the origin* $\psi^{\copyright} = O_{6\times 1}$ *results in the pose regulation* $g \to I_4$.

B. DISCRETIZATION BASED ON THE CAYLEY MAP

We now discretize the rigid body motion based on the Cayley parameter for $SE(3)$. With an analogy to (7), the kinematic equation of a rigid body can be written in the Cayley parametrization as follows:

$$
\psi_{k+1}^{\copyright} = \begin{bmatrix} \xi_{k+1}^{\copyright} \\ \eta_{k+1}^{\copyright} \end{bmatrix} = \mathrm{Cay}^{-1}(\mathrm{Cay}(\widehat{\psi}_k^{\copyright})\,\mathrm{Cay}(\widehat{V}_k^{\copyright} T))^{\vee}. \tag{13}
$$

This is a discrete-time kinematic equation with the Cayley parametrization, and it represents the same time evolution as (3). It is geometrically exact, as described for the exponential map in Fig. 3, provided that the twist $V^{\copyright}$ is constant for the time period. We show the detailed closed-form equation of (13), which is not in the literature. The direct calculation yields

$$
\begin{aligned}
\xi_{k+1}^{\copyright} &= \frac{I_3 - \omega_k^{\copyright} T \eta_k^{\copyright\top} - \widehat{\omega}_k^{\copyright} T}{1 - \eta_k^{\copyright\top}\omega_k^{\copyright} T}\xi_k^{\copyright} \\
&\quad + \frac{I_3 - \eta_k^{\copyright}\omega_k^{\copyright\top} T + \widehat{\eta}_k^{\copyright}}{1 - \eta_k^{\copyright\top}\omega_k^{\copyright} T} v_k^{\copyright} T,
\end{aligned} \tag{14}
$$

$$
\eta_{k+1}^{\copyright} = \frac{I_3}{1 - \eta_k^{\copyright\top}\omega_k^{\copyright} T}\eta_k^{\copyright} + \frac{I_3 + \widehat{\eta}_k^{\copyright}}{1 - \eta_k^{\copyright\top}\omega_k^{\copyright} T}\omega_k^{\copyright} T. \tag{15}
$$

The equations only require matrix and vector computation. Therefore, it is easy to compute closed forms of mathematical utilities such as the Jacobians of $\psi_{k+1}^{\copyright}$ and the gradient of a function with respect to trajectories defined by this formula. These are useful for solving an optimal control problem efficiently and will be used in Section IV.B.

We next consider the double-integrator system (2) on $SE(3)$. For simplicity, we neglect second- or higher order terms regarding the continuous change in the velocity of the body. This means that both the twist (linear and angular velocity) and the wrench (force and torque) do not change during a sampling period. Under this assumption, the whole discrete-time dynamical model of the double-integrator system on $SE(3)$ is expressed as follows:

$$
\psi_{k+1}^{\copyright} = \mathrm{Cay}^{-1}(\mathrm{Cay}(\widehat{\psi}_k^{\copyright})\,\mathrm{Cay}(\widehat{V}_k^{\copyright} T))^{\vee}, \tag{16}
$$

$$
V_{k+1}^{\copyright} = V_k^{\copyright} + (u_k^{\copyright} + u_{\mathrm{grav},k}^{\copyright})T. \tag{17}
$$

Here, the gravitational acceleration term $u_{\mathrm{grav},k}^{\copyright}$ is calculated by applying the transformation (11) to u_{grav}. The corresponding trajectory of the pose on the Lie group $SE(3)$ can be calculated from $\psi^{\copyright}$ through the Cayley map (8), and thus, the Lie group structure is always preserved.

Remark 4: The assumption on constant twist and wrench may cause small numerical errors in the predicted trajectory. As we use an NMPC scheme in this paper, however, the error can be treated as a disturbance affecting the twist, and it can be attenuated by feedback. If the higher order terms are needed, we can directly compute them by considering the Taylor series expansion of the Cayley map in the same manner as in [26]. We can also improve the accuracy of the trajectory by using shorter sampling periods in the transient state. It is a part of our future work.

Remark 5: The angular velocity parameter $\omega_k^{\copyright}$ *must satisfy* $\eta_k^{\copyright\top}\omega_k^{\copyright} T \neq 1$. *The equality* $\eta_k^{\copyright\top}\omega_k^{\copyright} T = 1$ *means that the orientation at the* $(k+1)$*th step becomes singular. For the case* $\eta_k^{\copyright\top}\omega_k^{\copyright} T > 1$ *where the orientation leaps across a singular orientation, the denominator* $1 - \eta_k^{\copyright\top}\omega_k^{\copyright} T$ *becomes negative. This might seem as if the rotation direction is inverted, but it does not affect the result of integration since rotation by the angles* θ *and* $\theta - 2\pi$ *gives the same orientation.*

IV. REAL-TIME NMPC OF RIGID BODY MOTION

A. PROBLEM SETTING

In this paper, we consider nonlinear model predictive control of the discrete-time rigid body dynamics (16) and (17). For simplicity, we use the same value T for the control period and the sampling time of the prediction horizon. We solve the following nonlinear programming problem at each control period in a receding horizon manner:

$$
J = \frac{1}{2}\sum_{k=1}^{N}\left(\|\psi_k^{\copyright}\|_{Q_p}^2 + \|V_k^{\copyright}\|_{Q_v}^2 + \|u_{k-1}^{\copyright} - u_{\mathrm{term}}^{\copyright}\|_{Q_u}^2\right),
$$

$$
\min_{\substack{\psi_1^{\copyright},\ldots,\psi_N^{\copyright}, \\ V_1^{\copyright},\ldots,V_N^{\copyright}, \\ u_0^{\copyright},\ldots,u_{N-1}^{\copyright}}} J \text{ subject to } (16), (17),\ u_{\mathrm{lb}}^{\copyright} \le u_k^{\copyright} \le u_{\mathrm{ub}}^{\copyright}. \tag{18}
$$

The objective function J is a quadratic form of the Cayley parameters, and box constraints on the acceleration input are considered. Here, we use the notation $\|x\|_A^2 = x^{\top}Ax$.

Q_p, Q_v, and Q_u are positive semidefinite 6×6 weight matrices for the pose, twist, and acceleration parameters, respectively. We assume that the initial states $\psi_0^{\copyright}$ and $V_0^{\copyright}$ are given, and terminal conditions are not specified. The control objective is to achieve $g_k \to I_4$ by $\psi_k^{\copyright} \to O_{6\times 1}$, $V_k^{\copyright} \to O_{6\times 1}$. In the objective function, $u_{\text{term}}^{\copyright} = [0\ 0\ \frac{G}{2}\ 0\ 0\ 0]^\top$ denotes the equilibrium input at the desired terminal state $g = I_4$, $V = O_{6\times 1}$. We consider the input cost $\frac{1}{2}\|u_{k-1}^{\copyright} - u_{\text{term}}^{\copyright}\|_{Q_u}^2$ to ensure that the incremental cost becomes zero at the desired terminal state and to induce $u^{\copyright} \to u_{\text{term}}^{\copyright}$.

We stack the state and input time series into vectors as follows:

$$\boldsymbol{\psi}^{\copyright} = \begin{bmatrix} \psi_1^{\copyright} \\ \psi_2^{\copyright} \\ \vdots \\ \psi_N^{\copyright} \end{bmatrix},\ \boldsymbol{V}^{\copyright} = \begin{bmatrix} V_1^{\copyright} \\ V_2^{\copyright} \\ \vdots \\ V_N^{\copyright} \end{bmatrix},\ \boldsymbol{u}^{\copyright} = \begin{bmatrix} u_0^{\copyright} - u_{\text{term}}^{\copyright} \\ u_1^{\copyright} - u_{\text{term}}^{\copyright} \\ \vdots \\ u_{N-1}^{\copyright} - u_{\text{term}}^{\copyright} \end{bmatrix}.$$

We now rewrite the objective function (18) using these time series vectors as

$$J = \frac{1}{2}\left(\boldsymbol{\psi}^{\copyright\top}\boldsymbol{Q}_p\boldsymbol{\psi}^{\copyright} + \boldsymbol{V}^{\copyright\top}\boldsymbol{Q}_v\boldsymbol{V}^{\copyright} + \boldsymbol{u}^{\copyright\top}\boldsymbol{Q}_u\boldsymbol{u}^{\copyright}\right). \tag{19}$$

Here, $\boldsymbol{Q}_p$, $\boldsymbol{Q}_v$, and $\boldsymbol{Q}_u$ are defined as

$$\boldsymbol{Q}_p = I_N \otimes Q_p,\ \boldsymbol{Q}_v = I_N \otimes Q_v,\ \boldsymbol{Q}_u = I_N \otimes Q_u,$$

where $\otimes$ denotes the Kronecker product such that

$$I_N \otimes Q_p = \text{blkdiag}(\underbrace{Q_p, \dots, Q_p}_{N}).$$

We use standard nonlinear optimization algorithms to minimize J, instead of using LGVI-based methods. This is because both the initial and terminal states are constrained in LGVI-based methods, and imposing some input constraints may result in ill-conditioned finite-time control problems. In contrast, the terminal state is not constrained in our approach, and thus such an issue merely arises as long as the constraints are well-conditioned.

B. REDUCED DECISION VARIABLES AND ANALYTIC GRADIENT COMPUTATION

We employ the recursive discretization technique [9] to eliminate the states $\psi_k^{\copyright}$ and $V_k^{\copyright}$ from (18). Recursively applying (16) and (17), we obtain

$$\begin{aligned} &V_{k+1}^{\copyright}(V_0^{\copyright}, \boldsymbol{u}^{\copyright}) \\ &\quad = V_0^{\copyright} + \sum_{j=0}^{k}(u_j^{\copyright} + u_{\text{grav},j}^{\copyright}(\psi_j^{\copyright}, V_j^{\copyright}))T, \end{aligned} \tag{20}$$

$$\begin{aligned} &\psi_{k+1}^{\copyright}(\psi_0^{\copyright}, V_0^{\copyright}, \boldsymbol{u}^{\copyright}) \\ &\quad = f(..f(f(\psi_0^{\copyright}, V_0^{\copyright}), V_1^{\copyright})\dots, V_k^{\copyright}), \end{aligned} \tag{21}$$

where $f(\psi_k^{\copyright}, V_k^{\copyright}) = \psi_{k+1}$. The objective function (18) is then rewritten as a function that takes only the initial conditions $\psi_0^{\copyright}$, $V_0^{\copyright}$ and the input sequence $\boldsymbol{u}^{\copyright}$. The resulting nonlinear optimization problem becomes as follows:

$$\min_{\boldsymbol{u}^{\copyright}} J(\psi_0^{\copyright}, V_0^{\copyright}, \boldsymbol{u}^{\copyright}) \quad \text{subject to } u_{\text{lb}}^{\copyright} \le u_k^{\copyright} \le u_{\text{ub}}^{\copyright}. \tag{22}$$

In this case, the decision variables of the NMPC problem are now only the input sequence $\boldsymbol{u}^{\copyright}$, and the number of them is reduced from $18N$ to $6N$. Moreover, the dynamics constraints (16) and (17) are no longer needed since they are satisfied during the calculation of (20) and (21).

Instead of using finite-difference approximations, we compute the closed-form gradient of the objective function (18) with respect to the input time series $\boldsymbol{u}^{\copyright}$ by utilizing (20) and (21). This is because centered finite-difference approximations evaluate the objective function $12N$ times, that is, twice the number of decision variables. They usually take more time than the analytic computation due to the repetitive recursive integration of (16) with (21).

From (19), the analytic gradient is readily available as follows:

$$\frac{\partial J}{\partial \boldsymbol{u}^{\copyright}} = \boldsymbol{\psi}^{\copyright\top}\boldsymbol{Q}_p\frac{\partial \boldsymbol{\psi}^{\copyright}}{\partial \boldsymbol{u}^{\copyright}} + \boldsymbol{V}^{\copyright\top}\boldsymbol{Q}_v\frac{\partial \boldsymbol{V}^{\copyright}}{\partial \boldsymbol{u}^{\copyright}} + \boldsymbol{u}^{\copyright\top}\boldsymbol{Q}_u.$$

The Jacobian matrices $\frac{\partial \boldsymbol{\psi}^{\copyright}}{\partial \boldsymbol{V}^{\copyright}}$ and $\frac{\partial \boldsymbol{V}^{\copyright}}{\partial \boldsymbol{u}^{\copyright}}$ are calculated as block lower triangular matrices in the following equations:

$$\frac{\partial \boldsymbol{\psi}^{\copyright}}{\partial \boldsymbol{u}^{\copyright}} = \begin{bmatrix} O_6 & \cdots & \cdots & O_6 \\ \frac{\partial \psi_2^{\copyright}}{\partial u_0^{\copyright}} & O_6 & \cdots & O_6 \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial \psi_N^{\copyright}}{\partial u_0^{\copyright}} & \cdots & \frac{\partial \psi_N^{\copyright}}{\partial u_{N-2}^{\copyright}} & O_6 \end{bmatrix}, \tag{23}$$

$$\frac{\partial \boldsymbol{V}^{\copyright}}{\partial \boldsymbol{u}^{\copyright}} = \begin{bmatrix} \frac{\partial V_1^{\copyright}}{\partial u_0^{\copyright}} & \cdots & \cdots & O_6 \\ \frac{\partial V_2^{\copyright}}{\partial u_0^{\copyright}} & \frac{\partial V_2^{\copyright}}{\partial u_1^{\copyright}} & \cdots & O_6 \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial V_N^{\copyright}}{\partial u_0^{\copyright}} & \cdots & \frac{\partial V_N^{\copyright}}{\partial u_{N-2}^{\copyright}} & \frac{\partial V_N^{\copyright}}{\partial u_{N-1}^{\copyright}} \end{bmatrix}. \tag{24}$$

From (20) and (21), we have recurrence formulas of the block components as follows:

$$\frac{\partial u_{\text{grav},n+1}^{\copyright}}{\partial u_m^{\copyright}} = \left.\frac{\partial u_{\text{grav}}^{\copyright}}{\partial \psi^{\copyright}}\right|_{\substack{\psi^{\copyright}=\psi_n^{\copyright} \\ V^{\copyright}=V_n^{\copyright}}} \cdot \frac{\partial \psi_n^{\copyright}}{\partial u_m^{\copyright}} + \left.\frac{\partial u_{\text{grav}}^{\copyright}}{\partial V^{\copyright}}\right|_{\substack{\psi^{\copyright}=\psi_n^{\copyright} \\ V^{\copyright}=V_n^{\copyright}}} \cdot \frac{\partial V_n^{\copyright}}{\partial u_m^{\copyright}},$$

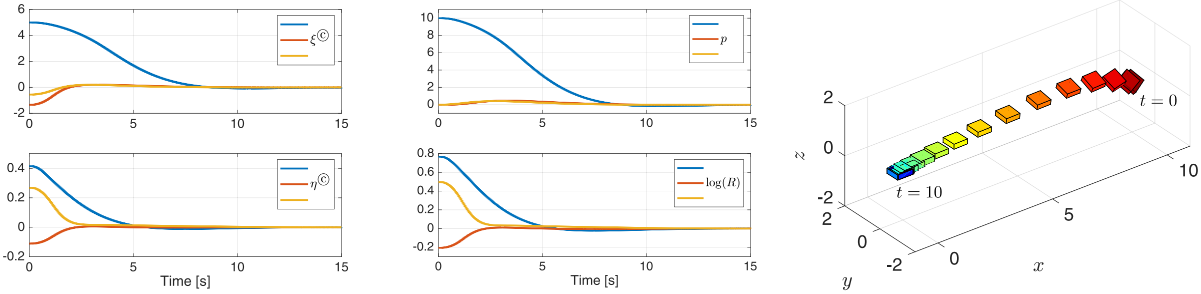
$$\frac{\partial V_{n+1}^{\copyright}}{\partial u_m^{\copyright}} = \frac{\partial V_n^{\copyright}}{\partial u_m^{\copyright}} + \frac{\partial u_{\text{grav},n}^{\copyright}}{\partial u_m^{\copyright}}T,$$

$$\frac{\partial \psi_{n+1}^{\copyright}}{\partial u_m^{\copyright}} = \left.\frac{\partial f}{\partial \psi^{\copyright}}\right|_{\substack{\psi^{\copyright}=\psi_n^{\copyright} \\ V^{\copyright}=V_n^{\copyright}}} \cdot \frac{\partial \psi_n^{\copyright}}{\partial u_m^{\copyright}} + \left.\frac{\partial f}{\partial V^{\copyright}}\right|_{\substack{\psi^{\copyright}=\psi_n^{\copyright} \\ V^{\copyright}=V_n^{\copyright}}} \cdot \frac{\partial V_n^{\copyright}}{\partial u_m^{\copyright}}$$

for $n \ge m+2,\ m \ge 0$, and

$$\frac{\partial u_{\text{grav},m+1}^{\copyright}}{\partial u_m^{\copyright}} = O_6,\ \frac{\partial V_{m+1}^{\copyright}}{\partial u_m^{\copyright}} = TI_6,\ \frac{\partial \psi_{m+1}^{\copyright}}{\partial u_m^{\copyright}} = O_6.$$

Here, the states $\psi_k^{\copyright}$ and $V_k^{\copyright}$ are precalculated by (20) and (21) and plugged into Jacobian matrices $\frac{\partial u_{\text{grav}}^{\copyright}}{\partial \psi^{\copyright}}$, $\frac{\partial u_{\text{grav}}^{\copyright}}{\partial V^{\copyright}}$, $\frac{\partial f}{\partial \psi^{\copyright}}$, and $\frac{\partial f}{\partial V^{\copyright}}$. Therefore, the gradient can be computed only with the initial state $\psi_0^{\copyright}$, $V_0^{\copyright}$ and the control input sequence $\boldsymbol{u}^{\copyright}$. Moreover, with the sparse structure of the Jacobian matrices



(a) Pose trajectory in the Cayley parametrization

(b) Pose trajectory

(c) Visualization of the trajectory

FIGURE 5. Time response of pose (Simulation A). The convergence of the Cayley parameter $\psi^{\copyright} = \left[\xi^{\copyright\top}\ \eta^{\copyright\top}\right]^{\top}$ implies the convergence of the pose $g = \mathrm{Cay}(\widehat{\psi}^{\copyright})$ to the identity. (a) Pose trajectory in the Cayley parametrization. (b) Pose trajectory. (c) Visualization of the trajectory.

$\frac{\partial \boldsymbol{\psi}^{\copyright}}{\partial \boldsymbol{u}^{\copyright}}$ and $\frac{\partial \boldsymbol{V}^{\copyright}}{\partial \boldsymbol{u}^{\copyright}}$, the total computation time for matrix multiplication can be reduced to less than half of that for dense matrix multiplication.

C. ACCELERATION INPUT CONSTRAINT

The input constraints should be imposed on the physical acceleration u, but the control input is its corresponding Cayley parameter $u^{\copyright}$, which has no physical meaning. As we have shown in (11) and (12), the transformation between u and $u^{\copyright}$ requires the twist $V = [v^{\top}\ \omega^{\top}]^{\top}$ which is a part of the state. This means that the simple box constraints on the actual acceleration u turn into complicated nonlinear constraints on $u^{\copyright}$, and this is undesirable.

Although we cannot strictly constrain the actual acceleration, we can approximate the input constraints when the angular velocity ω is sufficiently small. Here, we use the relation between u and $u^{\copyright}$ in (11) and (12) for $\|\omega\| = 0$. The approximated lower and upper bounds u_{lb} and u_{ub} of the actual acceleration become as follows:

$$\tilde{u}_{\mathrm{lb}} \approx 2u^{\copyright}_{\mathrm{lb}},\ \tilde{u}_{\mathrm{ub}} \approx 2u^{\copyright}_{\mathrm{ub}}.$$

As we will see in the next section, this approximation is computationally efficient and behaves well in both the transient and steady states.

V. ON-BOARD SIMULATION

We have implemented the NMPC algorithm on a Raspberry Pi 3 Model B+ single-board computer (ARMv8 CPU, 1.4 GHz) in C++. We have used the low-storage Broyden-Fletcher-Goldfarb-Shanno algorithm [27] with box constraints (L-BFGS-B algorithm) implemented in NLopt [28] library to solve the problem (22). The dynamics of a rigid body are simulated by using the discretized model (14), (15), and (17), and the nonlinear programming problem (22) is solved at each step. We exploit the sparsity of Jacobian matrices (23) and (24) to compute the analytic gradient efficiently.

In this section, we show numerical examples for the following two applications:

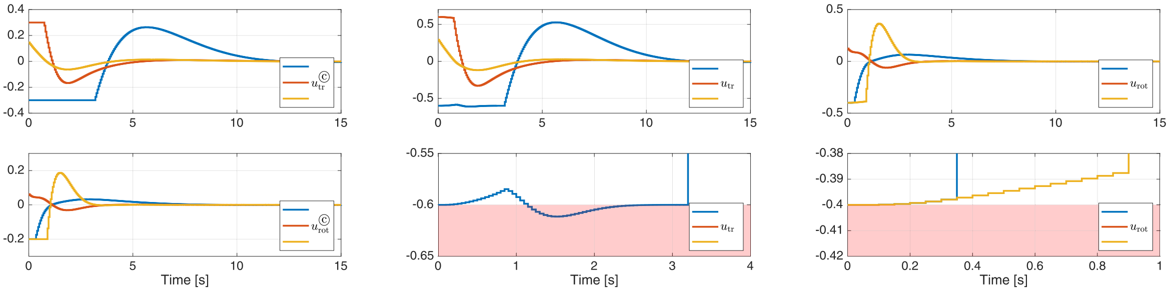
- Simulation A: Spacecraft with input saturation.
  The dynamical model (17) and the objective function (18) are calculated without gravity, i.e., $G = 0\,\mathrm{m\,s^{-2}}$. The acceleration input is constrained by using the approximation shown in the last section.
- Simulation B: A fully actuated aerial vehicle subject to gravity.
  The dynamical model and the objective function are calculated with gravitational acceleration constant $G = 9.8\,\mathrm{m\,s^{-2}}$. The control input is not constrained.

Both systems can be modeled as second-order fully actuated systems on $SE(3)$ [29], [30]. For both cases, the weight matrices in the objective function are $Q_p = Q_v = Q_u = I_6$. The initial state is given by $\psi^{\copyright}_0 = \mathrm{Cay}^{-1}(g_0)$ and $V^{\copyright}_0 = O_{6\times1}$, where $g_0 \in SE(3)$ is the initial pose. The control objective is to achieve $g_k \to I_4$ by $\psi^{\copyright}_k \to O_{6\times1}$, $V^{\copyright}_k \to O_{6\times1}$.

A. SIMULATION A

In Simulation A, we consider optimal pose regulation of spacecraft with input saturation. This can also be identified with a dynamical system of a fully actuated aerial vehicle with gravity compensation applied. The input is constrained by $u^{\copyright}_{\mathrm{ub}} = [0.3\ 0.3\ 0.3\ 0.2\ 0.2\ 0.2]^{\top}$ and $u^{\copyright}_{\mathrm{lb}} = -u^{\copyright}_{\mathrm{ub}}$, that is, $u_{\mathrm{ub}} \approx [0.6\ 0.6\ 0.6\ 0.4\ 0.4\ 0.4]^{\top}$. The gravitational acceleration constant is set to $G = 0$, and accordingly, $u^{\copyright}_{\mathrm{term}} = u^{\copyright}_{\mathrm{grav},k} = O_{6\times1}$. The initial pose is defined as $g_0 = (R_0, p_0)$, where $R_0 = R_x(\frac{\pi}{4})R_z(\frac{\pi}{6})$ and $p_0 = [10\ 0\ 0]^{\top}$. The duration of the prediction horizon is 1 s, and it is divided into $N = 20$ periods (i.e., $T = 0.05$ s).

Figure 5 shows the optimal trajectory of the pose g and the corresponding Cayley parameter $\psi^{\copyright}$. The acceleration input in the Cayley parametrization $u^{\copyright}$ is shown in Fig. 6(a). The pose $g = \mathrm{Cay}(\widehat{\psi}^{\copyright})$ converges to the identity under the input constraints. The physical acceleration u calculated from the corresponding Cayley parameter $u^{\copyright}$ is shown in Fig. 6. Although the approximated lower bound $\tilde{u}_{\mathrm{lb}}$ is violated at some steps, as illustrated in Fig. 6(b), the error is sufficiently small and is not critical in practice. The measured computation time for optimization is shown in Fig. 7. The computation time is far less than the sampling period, and the present method successfully achieves real-time computation of NMPC on board.



(a) Control input
(b) Linear acceleration u_{tr}
(c) Angular acceleration u_{rot}

FIGURE 6. Time response of input (Simulation A). Note that the control input $u^{\copyright} = \left[u_{\mathrm{tr}}^{\copyright\top}\ u_{\mathrm{rot}}^{\copyright\top}\right]^{\top}$ in (a) does not represent the physical acceleration. In (b), the linear acceleration u_{tr} violates the approximated constraint shown in the red region. However, the approximation error is sufficiently small and not critical in practice. (a) Control input.

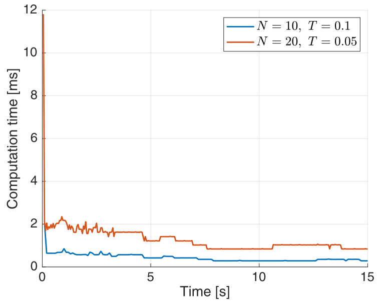


FIGURE 7. Computation time for each sampling period (Simulation A). The proposed NMPC method is feasible in real time on a Raspberry Pi single-board computer.

B. SIMULATION B

In Simulation B, we consider control of a fully actuated aerial vehicle without prior gravity compensation. The gravitational acceleration constant is set to $G = 9.8\,\mathrm{m\,s^{-2}}$. From (2) and (11), the equilibrium input at the desired terminal state is then given as $u_{\mathrm{term}}^{\copyright} = [0\ 0\ 4.9\ 0\ 0\ 0]^{\top}$. When the control input is only applied to counter the force of gravity, the input cost equals $\frac{1}{2}\|u_{\mathrm{grav},k}^{\copyright} + u_{\mathrm{term}}^{\copyright}\|_{Q_u}^2$. This value is positive except for $\psi_k^{\copyright} = O_6$, $V_k = O_6$, and we can say that the cost for gravity compensation is also included in the objective function (18). The initial pose is $R_0 = I_3$, $p_0 = [10\ 6\ 0]^{\top}$, and the acceleration input is not constrained in this simulation. The prediction horizon duration is 2 s, and it is divided into $N = 20$ periods (i.e., $T = 0.1$ s). We have selected the prediction horizon longer than that of Simulation A to improve stability.

Figure 8 shows the optimal pose trajectory and control input. The rigid body decelerates by tilting the body in the opposite direction of travel. This notable behavior is the result of considering the gravity compensation cost. The sum of squared norm of $u_0^{\copyright} - u_{\mathrm{term}}^{\copyright}$ is 46.03 when we utilize the body tilt with the acceleration input in the $+z$-direction of the body frame. If we force the controller to use only the horizontal acceleration by constraining the rotational input $u_{\mathrm{rot}}^{\copyright}$ to be always zero, the value becomes 74.81. This result shows that the most efficient maneuver subject to gravity is not the parallel motion. It is also shown that our present NMPC method and the objective function can deal with the external force, which is difficult to consider in analytic solution methods.

TABLE 1. Summary of the computation time on a Raspberry Pi embedded computer.

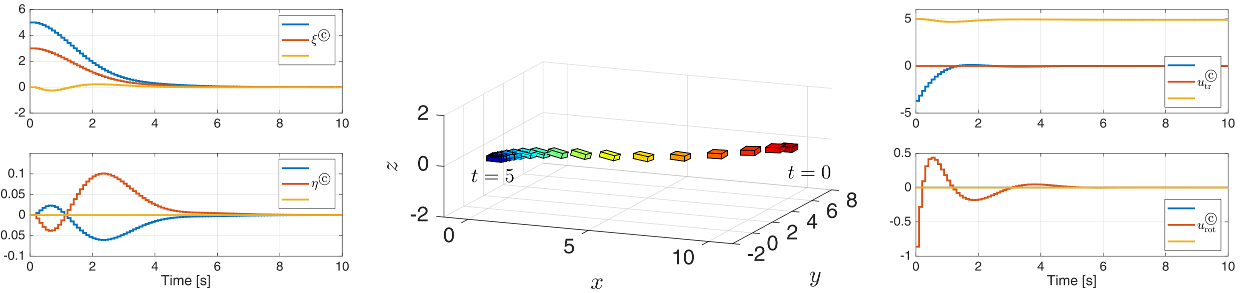
Settings	Min. [ms]	Med. [ms]	Max. [ms]	Real-time?
Simulation A $N = 10$ $T = 100$ ms	0.28	0.29	2.05	Y
Simulation A $N = 20$ $T = 50$ ms	0.84	1.04	11.81	Y
Simulation A $N = 20$ $T = 100$ ms	0.16	1.52	14.77	Y
Simulation A $N = 40$ $T = 100$ ms	1.11	8.87	130.63	N
Simulation B $N = 20$ $T = 50$ ms	7.22	9.35	16.98	Y
Simulation B $N = 20$ $T = 100$ ms	4.79	13.12	49.89	Y
Simulation B $N = 40$ $T = 100$ ms	17.24	101.77	460.26	N

The measured computation time for the optimization is shown in Fig. 9. The optimization takes longer time than that of Simulation A because computation regarding $u_{\mathrm{grav}}^{\copyright}$ is additionally required in this case. It is still applicable to real-time applications since the computation time remains less than half of the control period.

C. DISCUSSION

Note that long computation time is observed in the first period of Simulations A and B. We can avoid this by setting a proper initial guess of the input sequence. We have experimented for several other cases with different N and T to determine how they affect the computation time. The computation time is summarized in Table 1. As a result, the longer prediction horizon leads to longer computation, and the required time is affected not only by the number of prediction steps N but also by the sampling period T.

As seen in Table 1, we cannot afford $N = 40$ in this case, since the computation time significantly increases with increasing N. To achieve real-time computation for a long prediction horizon together with a fine time resolution, we may need to combine our proposed techniques with more efficient NMPC algorithms such as the continuation/GMRES method [31]. We can also consider nonuniform sampling time



(a) Pose trajectory (b) Visualization of the trajectory (c) Control input

FIGURE 8. Time response of pose and input (Simulation B). The rigid body decelerates by tilting the body in the opposite direction of travel. This behavior is achieved by considering the cost for gravity compensation. (a) Pose trajectory. (b) Visualization of the trajectory. (c) Control input.

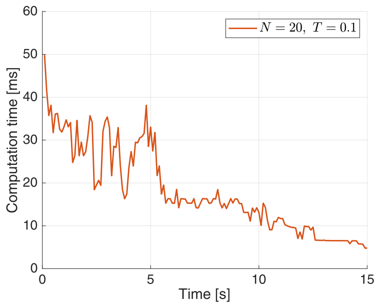


FIGURE 9. Computation time for each sampling period (Simulation B). The computational time is longer than that of Simulation A, but it is still feasible in real time.

as in [32] and [33] to use a fine sampling time resolution in the transient state while sampling coarsely in the steady state, which can achieve a long horizon with a small N. Since our discretization method based on the Cayley map is geometrically exact, it is easy to apply it to the nonuniform-sampling NMPC without being concerned about the variable numerical precision due to approximations. Such additional techniques could be used to further improve the computational performance.

VI. EXPERIMENT

In this section, we apply the proposed NMPC method to a fully actuated hexarotor system and conduct a pose control experiment. The objective of the experiment is to show the real-time feasibility of the proposed method and with some input constraints and the force of gravity considered.

Figure 1 is a picture of the experimental vehicle. The rotors, which consists of motors and propellers, are tilted and fixed so that the vehicle can generate force and torque in every direction and around any axis. The dynamics of this vehicle can be modeled as a fully actuated second-order system on $SE(3)$ [4]. The mass and inertia tensor of the vehicle are $m = 0.61$ kg and $\mathcal{I} = \mathrm{diag}(2.5 \times 10^{-3}, 2.5 \times 10^{-3}, 4.6 \times 10^{-3})$ kg m^2, respectively. Fig. 10 shows the block diagram of the experimental system. Controllers are implemented on the Raspberry Pi used in the last section and a 3D Robotics Pixhawk Mini flight controller by using Simulink and Embedded Coder. The high-level controller running on the Raspberry Pi estimates the current state and calculates the NMPC input at 20 Hz ($T = 0.05$ s). The computed NMPC

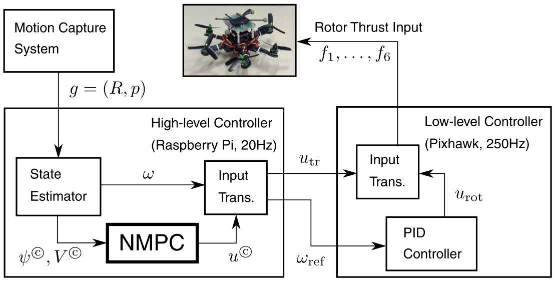
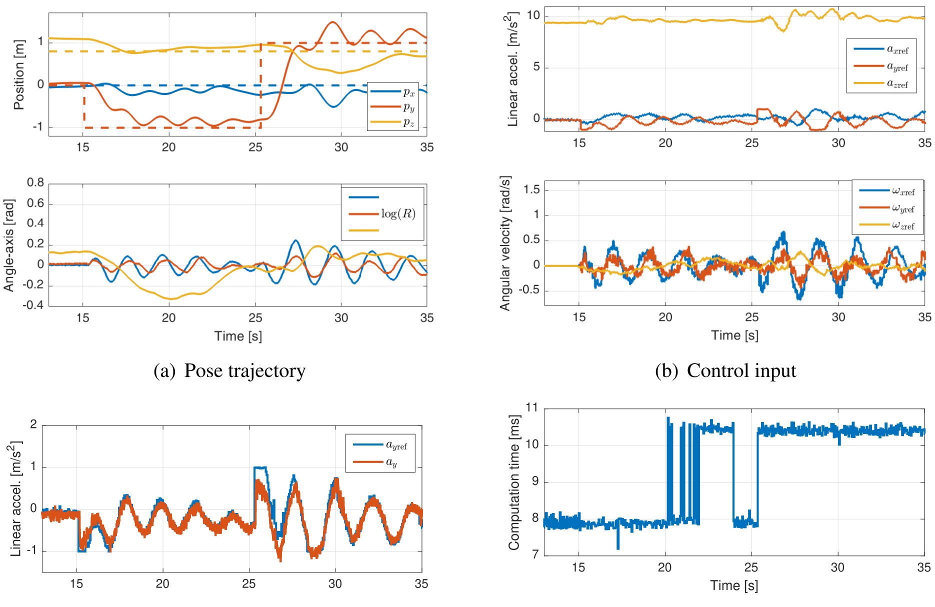


FIGURE 10. Block diagram of the experimental system. The present NMPC method is implemented on the high-level controller.

input $u^{\copyright}$ is converted to the real acceleration vector u by using the inverse maps of (11) and (12). The Pixhawk Mini implements the low-level controller, which controls the linear acceleration $u_{\mathrm{tr}} = \begin{bmatrix} a_{x\mathrm{ref}} & a_{y\mathrm{ref}} & a_{z\mathrm{ref}} \end{bmatrix}^{\top}$ and the angular velocity ω by a feed-forward and PID controller, respectively.

The pose of the vehicle is measured by an OptiTrack motion capture system by observing the reflective markers on the body, and the information is sent to the high-level controller via UDP on the wireless LAN at the rate 100 Hz. The pose information from the motion capture system to the vehicle has a latency of approximately 0.1 s. Due to this latency and the relatively long control period of NMPC, the attitude dynamics easily become unstable when we use angular acceleration as the control input. It is especially critical if modeling errors of force and torque exist, which can include unmodeled rotor dynamics or the aerodynamic behavior such as the ground effect. As the verification of the real-time NMPC is the objective of this experiment, we do not directly command the angular acceleration as the input. Instead, we control the angular velocity with the low-level controller. The angular velocity input is calculated by (17) with the estimated current state and the NMPC input.

In this experiment, the vehicle is first commanded to stay at the initial pose $(R, p) = (I_3, [0\ 0\ 0.8]^{\top})$, and then at $t \approx 15$ s and 25 s, the reference y coordinate is changed to -1 and 1, respectively. The reference position is set by shifting the origin. The origins of the attitude and twist are $R = I_3$ and $V = O_{6\times1}$, respectively. The objective function is the same as that of simulations defined in (19). We set the weight matrix for the pose to $Q_p = 5I_6$ in order to penalize the position error, and $Q_v = Q_u = I_6$ for the twist and the



(a) Pose trajectory

(b) Control input

(c) Response of acceleration

(d) Computation time

FIGURE 11. **Result of the experiment with a fully actuated hexarotor. The pose trajectory and the control input are displayed in (a) and (b), respectively. (c) shows that the vehicle can quickly respond to the acceleration input. The computation time for NMPC shown in (d) confirms that the present NMPC method is also real-time feasible in the experiment. (a) Pose trajectory. (b) Control input. (c) Response of acceleration. (d) Computation time.**

control input. According to the simulation results, the prediction horizon is defined by $N = 20$ and $T = 0.05$ s. The gravitational acceleration constant is set to $G = 9.8\,\mathrm{m\,s^{-2}}$ to explicitly compensate for the force of gravity. We constrain the horizontal linear acceleration $a_{x\text{ref}}$ and $a_{y\text{ref}}$ in $[-1, 1]$ by restricting the corresponding Cayley parameters in $u_{tr}^{\copyright}$ to $[-0.5, 0.5]$.

Figure 11 shows the result of this experiment. The pose trajectory and the control input are displayed in Figs. 11(a) and 11(b), respectively. The pose successfully converges to the step reference in the y coordinate. In Fig. 11(c), the time response of the y-acceleration to the reference value is presented. Although the transient error is observed, we can see that the vehicle immediately responds to the change in reference acceleration. We can also confirm that the acceleration input constraint in y axis is activated from $t \approx 15$ to 15.5 and from $t \approx 25$ to 26. The computation time for NMPC shown in Fig. 11(d) remains at approximately 0.01 s during the experiment. This time is sufficiently shorter than the control period 0.05 s, and the result verifies that the present NMPC method is also real-time feasible in the experimental setup with a low-cost single-board computer. This result is notable because there are only a few studies that report experiments of the real-time mobile robot control by on-board NMPC computation. By combining the exact discretization method and the recursive discretization technique, we have achieved reduction of the decision variables and the fast analytic computation of the objective function gradient.

The experiment shows local stability in presence of disturbances such as unmodeled dynamics of the battery or rotors. Although the stability is not proven yet, the stabilizing constraint technique discussed in [9, Chapter 5] could be applied to our case. The main challenge is that we need to explicitly consider the external force, namely the force of gravity, in the stability analysis.

VII. CONCLUSION

In this paper, we have presented a fast NMPC method for rigid body dynamics that can run on an embedded computer in real time. We have combined the following methods to speed up the nonlinear optimization:

- Geometrically exact discretization based on the Cayley map: it allows us to pick coarse sampling intervals and reduces the required number of prediction steps.
- The recursive discretization technique [9]: it reduces the number of decision variables by eliminating the sequence of states from the objective function and constraints.
- The analytic gradient of the objective function: with the Cayley-map-based discretization, the gradient is easily calculated, and the sparsity of intermediate matrices can be exploited to speed up matrix multiplication.

The present NMPC method is feasible in real time on a Raspberry Pi single-board computer, and it is shown by simulations and an experiment with a fully actuated hexarotor.

Future work includes NMPC with a singularity-free representation of the rigid body pose, the stability analysis, nonlinear state constraints with reduced computational effort, and nonuniform sampling intervals to consider a long prediction horizon with a small number of prediction steps.

REFERENCES

[1] F. Bullo and R. Murray, “Proportional derivative (PD) control on the Euclidean group,” in *Proc. 3rd Eur. Control Conf.*, 1995, pp. 1091–1097.

[2] T. Lee, M. Leok, and N. H. Mcclamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *Proc. 49th IEEE Conf. Decis. Control (CDC)*, Dec. 2010, pp. 5420–5425.

[3] C. Liu, S. Tang, and J. Guo, "Intrinsic optimal control for mechanical systems on lie group," *Adv. Math. Phys.*, vol. 2017, pp. 1–8, 2017.

[4] Y. Tadokoro, T. Ibuki, and M. Sampei, "Joint optimization of geometric control and structure of a fully-actuated hexrotor based on an analytic HJBE solution," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 1186–1191.

[5] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, 2nd ed. New York, NY, USA: Springer-Verlag, 2006.

[6] T. Lee, M. Leok, and N. H. Mcclamroch, "Computational geometric optimal control of rigid bodies," *Commun. Inf. Syst.*, vol. 8, no. 4, pp. 445–472, 2008.

[7] M. B. Kobilarov and J. E. Marsden, "Discrete geometric optimal control on lie groups," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 641–655, Aug. 2011.

[8] M. Kobilarov, "Discrete optimal control on lie groups and applications to robotic vehicles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 5523–5529.

[9] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. New York, NY, USA: Springer-Verlag, 2011.

[10] J. Selig, "Cayley maps for SE(3)," in *Proc. 12th IFToMM World Congr.*, 2007.

[11] M. Mehndiratta and E. Kayacan, "Reconfigurable fault-tolerant NMPC for Y6 coaxial tricopter with complete loss of one rotor," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2018, pp. 774–780.

[12] R. Quirynen, K. Berntorp, and S. Di Cairano, "Embedded optimization algorithms for steering in autonomous vehicles based on nonlinear model predictive control," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 3251–3256.

[13] I. K. Erunsal, A. Martinoli, and R. Ventura, "Decentralized nonlinear model predictive control for 3D formation of multirotor micro aerial vehicles with relative sensing and estimation," in *Proc. Int. Symp. Multi-Robot Multi-Agent Syst. (MRS)*, Aug. 2019, pp. 176–178.

[14] E. Rossi, M. Bruschetta, R. Carli, Y. Chen, and M. Farina, "Online nonlinear model predictive control for tethered UAVs to perform a safe and constrained maneuver," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 3996–4001.

[15] J. Dentler, "Real-time model predictive control of cooperative aerial manipulation," Ph.D. dissertation, Univ. Luxembourg, Luxembourg, 2018.

[16] M. Kamel, M. Burri, and R. Siegwart, "Linear vs nonlinear MPC for trajectory tracking applied to rotary wing micro aerial vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3463–3469, Jul. 2017.

[17] D. Bicego, J. Mazzetto, R. Carli, M. Farina, and A. Franchi, "Nonlinear model predictive control with actuator constraints for multi-rotor aerial vehicles," 2019, *arXiv:1911.08183*. [Online]. Available: https://arxiv.org/abs/1911.08183

[18] G. Garimella and M. Kobilarov, "Towards model-predictive control for aerial pick-and-place," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 4692–4697.

[19] G. Garimella, M. Sheckells, and M. Kobilarov, "Robust obstacle avoidance for aerial platforms using adaptive model predictive control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5876–5882.

[20] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL, USA: CRC Press, 1994.

[21] J. Gallier and D. Xu, "Computing exponentials of skew symmetric matrices and logarithms of orthogonal matrices," *Int. J. Robot. Autom.*, vol. 18, no. 1, pp. 10–20, 2003.

[22] J. Selig, *Geometrical Methods in Robotics*. New York, NY, USA: Springer-Verlag, 1996.

[23] D. Condurache and I. Ciureanu, "Closed form of the baker-Campbell-Hausdorff formula for the lie algebra of rigid body displacements," in *Proc. 9th ECCOMAS Thematic Conf. Multibody Dyn.*, 2019, pp. 307–314.

[24] H. Schaub, P. Tsiotras, and J. L. Junkins, "Principal rotation representations of proper N×N orthogonal matrices," *Int. J. Eng. Sci.*, vol. 33, no. 15, pp. 2277–2295, Dec. 1995.

[25] J. Dooley and J. Mccarthy, "On the geometric analysis of optimum trajectories for cooperating robots using dual quaternion coordinates," in *Proc. IEEE Int. Conf. Robot. Autom.*, Dec. 2002, pp. 1031–1036.

[26] J. Solà, "Quaternion kinematics for the error-state Kalman filter," 2017, *arXiv:1711.02508*. [Online]. Available: http://arxiv.org/abs/1711.02508

[27] J. Nocedal, "Updating quasi-Newton matrices with limited storage," *Math. Comput.*, vol. 35, no. 151, pp. 773–782, Sep. 1980.

[28] S. Johnson. *The NLopt Nonlinear-Optimization Package*. Accessed: Sep. 25, 2019. [Online]. Available: http://github.com/stevengj/nlopt

[29] Y. Tadokoro, T. Ibuki, and M. Sampei, "Classification and structural evaluation of fully-actuated hexrotor UAVs," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 1945–1950.

[30] T. Lee, N. H. Mcclamroch, and M. Leok, "Optimal control of a rigid body using geometrically exact computations on SE(3)," in *Proc. 45th IEEE Conf. Decis. Control*, 2006, pp. 2710–2715.

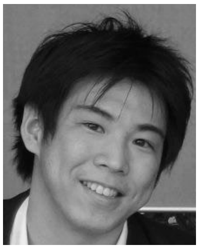
[31] T. Ohtsuka, "A continuation/GMRES method for fast computation of nonlinear receding horizon control," *Automatica*, vol. 40, no. 4, pp. 563–574, Apr. 2004.

[32] Y. Cao and W.-H. Chen, "Variable sampling-time nonlinear model predictive control of satellites using magneto-torquers," *Syst. Sci. Control Eng.*, vol. 2, no. 1, pp. 593–601, Dec. 2014.

[33] O. Gomozov, J. P. F. Trovao, X. Kestelyn, and M. R. Dubois, "Adaptive energy management system based on a real-time model predictive control with nonuniform sampling time for multiple energy storage electric vehicle," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 5520–5530, Jul. 2017.

YUICHI TADOKORO (Member, IEEE) received the B.Eng. and M.Eng. degrees in control engineering from the Tokyo Institute of Technology, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in systems and control engineering. Since 2018, he has been a Research Fellow with the Japan Society for the Promotion of Science. His research interests include control and design of aerial vehicles, mechatronics, nonlinear optimal control, and geometric methods for nonlinear control.

YUKI TAYA received the B.Eng. degree in control engineering from the Tokyo Institute of Technology, in 2018. He is currently with the Graduate School, Tokyo Institute of Technology. His research interests include robotics and mechatronics, control of aerial vehicles, and nonlinear control applications.



TATSUYA IBUKI received the B.Eng., M.Eng., and Ph.D.Eng. degrees from the Tokyo Institute of Technology, Japan, in 2008, 2010, and 2013, respectively. He was a Research Fellow with the Japan Society for the Promotion of Science from 2012 to 2013, and a Visiting Scholar with the School of Electrical and Computer Engineering, Georgia Institute of Technology, in 2019. He is currently an Assistant Professor with the Department of Systems and Control Engineering, Tokyo Institute of Technology. His research interests include cooperative control of robotic networks, vision-based estimation and control, and nonlinear control theory and its applications.

MITSUJI SAMPEI received the B.Eng., M.Eng., and Dr.Eng. degrees in control engineering from the Tokyo Institute of Technology, in 1983, 1985, and 1987, respectively. From 1987 to 1993, he was a Research Associate and an Associate Professor with Chiba University. Since 1993, he has been with the Tokyo Institute of Technology, where he is currently a Professor with the Department of Systems and Control Engineering. His current research interests are in the field of nonlinear system theory, including underactuated systems, nonholonomic systems, robotics, and H-infinity control theory. He served as an Organizing Committee Member of several conferences, including an IPC Chair of the IEEE Conference on Control Applications (CCA, 2010), of the SICE Annual Conference (SICE 2011), and of the IEEE Conference on Decision and Control (CDC 2015).