Nonlinear Dynamics manuscript No.
(will be inserted by the editor)

On the integration of singularity-free representations of $SO(3)$ for direct optimal control

Silvia Manara · **Marco Gabiccini** · **Alessio Artoni** · **Moritz Diehl**

Received: date / Accepted: date

Abstract In this paper we analyze the performance of different combinations of: (i) parameterization of the rotational degrees of freedom (DOF) of multibody systems, and (ii) choice of the integration scheme, in the context of direct optimal control discretized according to the direct multiple-shooting method. The considered representations include quaternions and Direction Cosine Matrices, both having the peculiarity of being non-singular and requiring more than three parameters to describe an element of the Special Orthogonal group $SO(3)$. These representations yield invariants in the dynamics of the system, i.e. algebraic conditions which have to be satisfied in order for the model to be representative of physical reality. The investigated integration schemes include the classical explicit Runge-Kutta method, its stabilized version based on Baumgarte's technique, which tends to reduce the drift from the underlying manifold, and the structure-preserving alternative, namely the Runge-Kutta Munthe-Kaas method, which preserves the invariants by construction. The performances of the combined choice of representation and integrator are assessed by solving thousands of planning tasks for a nonholonomic, underactuated cart-pendulum system, where the pendulum can experience arbitrarily large 3D rotations. The aspects analyzed include: success rate, average number of iterations and CPU time to convergence, and quality of the solution. The results reveal how structure-preserving integrators are the only choice for lower accuracies, whereas higher-order, non-stabilized standard integrators seem to be the computationally most competitive solution when higher levels of accuracy are pursued. Overall, the quaternion-based representation is the most efficient in terms of both iterations and CPU time to convergence, albeit at the cost of lower success rates and increased probability of being trapped by higher local minima.

Keywords Special Orthogonal group $SO(3)$ · Rotation parameterization · Direction Cosine Matrix · Quaternions · Lie group integrators · Direct optimal control · Spherical pendulum · Unicycle

S. Manara · M. Gabiccini · A. Artoni
Dipartimento di Ingegneria Civile e Industriale, University of Pisa, Largo Lucio Lazzarino 2, 56122 Pisa, Italy
Tel.: +39-050-2218011
Fax: +39-050-2218065
E-mail: silviamanara1@gmail.com

M. Diehl
Systems Control and Optimization Laboratory, Department of Microsystems Engineering (IMTEK), University of Freiburg, Georges-Köhler-Allee 102, 79110 Freiburg, Germany

1 Introduction

Numerical integrators preserving the structure of the nonlinear manifold on which the differential equation to be integrated is naturally evolving are well known to the scientific community [1]. These schemes have already been successfully applied to the simulation of multibody systems (MBS) [2, 3], to mention only a few.

The main reason why these integration techniques lend themselves to the simulation of the dynamics of systems of rigid bodies in a 3D space is proper handling of the rotational degrees of freedom (DOF). In fact, the orientation of a body in space is not a vectorial quantity: it rather belongs to a manifold, namely the Special Orthogonal group $SO(3)$, which can be parameterized in several different ways [4]. Since this manifold is *locally* isomorphic to $\mathbb{E}^3$, the three-dimensional Euclidean space, it can be *only locally* parameterized by, e.g., Euler angles, a chart of three independent coordinates. However, this minimal representation is not tailored to be employed in contexts where the orientation to be described spans a wide range. Euler angles suffer indeed

from the inevitable presence of representation singularities. In order to avoid this problem, often referred to as gimbal lock, non-minimal representations of $SO(3)$ must be employed, see [5] for an exhaustive overview. Non-minimal parameterizations overcome singularities by employing a higher-dimensional set of coordinates in the representation, such as the four parameters of a unit quaternion (Q), or the nine parameters corresponding to the elements of a rotation matrix. We will refer to the latter parameterization as Direction Cosine Matrix (DCM). The introduced redundant coordinates must satisfy additional algebraic constraints in order for the representation to consistently describe physical rigid-body rotations. Specifically, a quaternion is required to have unit length, whereas a DCM is required to be an orthogonal matrix. Since non-singular parameterizations evolve on nonlinear manifolds themselves (respectively the Special Unitary group $SU(2)$ for the unit quaternion and $SO(3)$ itself for the DCM parameterization [6]), standard schemes are not tailored to be employed in their numerical integration. Because of numerical errors, in fact, these objects tend to drift away from the manifold they belong to.

This issue is relevant also in the context of direct optimal control, as standard integrators may cause the solution of the optimal control problem to be inconsistent for long simulation times. This problem has already been addressed in [7], where the authors proposed to apply numerical techniques, such as Baumgarte's method [8], to stabilize the numerical integration on the Lie group. Despite being easy to implement, this method requires the choice of coefficients whose *a priori* calibration is not straightforward and whose values are not easy to relate to the size of the resulting integration error. A suitable alternative seems to be represented by the application of structure-preserving integration schemes [9] in the propagation of Lie group elements over time. For an overview on the application of such integrators to MBS, we refer to [10, 11, 12]. Different Lie group integration schemes are available in the literature [13, 14]. In particular, the Runge-Kutta Munthe-Kaas (RK-MK) scheme [15] consists in numerically integrating the differential equation on the Lie algebra, which is a linear space, through conventional Runge-Kutta (RK) methods, and then updating the element of the nonlinear manifold by means of the exponential mapping, which relates the Lie algebra to its group. This method has been widely exploited in the integration of rotations on $SO(3)$ [16]. In a recent contribution [17], an original integration scheme has been proposed which operates directly on the nonlinear manifold of unit quaternions by first integrating the dynamics of the rotational DOF on the Lie algebra $so(3)$, and then mapping the incremental rotation vector on $SU(2)$.

In this paper, we analyze the performance of such kind of Lie group integrators, respectively acting on $SU(2)$ and on $SO(3)$, in the numerical solution of an optimal control problem, which we tackle by employing a direct method [18]. In [19], the performance of non-conventional energy-preserving integration schemes in the context of direct optimal control was assessed. Here, instead, we are interested in comparing classical and geometric integrators that preserve, the former via stabilization, the latter by construction, the Lie group structure of the underlying singularity-free 3D rotation representation. To the best of our knowledge, this is the first time that a comparison between structure-preserving and standard integrators has been assessed in the context of numerical optimal control. To this end, RK-MK integrators are exploited in the solution of a trajectory optimization for a simple but challenging benchmark mechanical system. Namely, it is a two-wheeled spherical pendulum, which is made up of a unicycle connected to a pendulum by a passive spherical joint (Fig. 1).

The problem of controlling an inverted pendulum has traditionally attracted significant interest in nonlinear control literature because of the nonlinearity and instability of its dynamics. Several control strategies have been proposed to deal with the stabilization of the spherical pendulum around its unstable equilibrium position, such as [20, 21]. Interesting contributions were also devoted to study the swing-up problem. For instance, in [22] the problem of steering the spherical joint to a given position while swinging up the pendulum is addressed. Here, the pendulum is actuated by a manipulator capable of delivering any 3D acceleration to the spherical joint, within the limits of the control torques. In [23] a controller is proposed that can bring the pendulum to the upright position, starting from any orientation in the upper hemisphere. The spherical joint is actuated by an external force, which can have any direction on the plane the joint is enforced to belong to.

Our trajectory planning problem is made significantly more difficult by the nonholonomic constraints due to the presence of the unicycle on which the spherical pendulum is installed. These constraints, in fact, limit the instantaneous acceleration of the spherical joint to the longitudinal direction of the cart. Even the control of simpler nonholonomic vehicles, where the inverted pendulum is connected to a two-wheeled cart by a cylindrical joint, still seems to be an open research issue: several recently published papers [24, 25, 26] were devoted to tackle this problem with different nonlinear control tools.

The system we consider in this paper has been studied in a recent contribution [27]. However, our goal here is significantly different: while in such paper the focus is on the linearization-based design of a controller aimed at stabilizing the inverted pendulum in a neighborhood of its upright position (by maneuvering the cart), our aim is to find optimal trajectories to steer the unicycle to a given (possibly distant) position *while* swinging the pendulum from the low to the upright position. Because of the wideness of the do-

main of pendulum orientations to be spanned, the rotation parameterization based on Euler angles is unsuitable for this application, and a singularity-free representation needs to be employed.

2 Dynamic model

The benchmark system of the present study is represented in Fig. 1. It consists of a unicycle, whose two wheels are separately actuated by two independent torques, and of a pendulum connected to its axle through a passive spherical joint. The values of geometric and inertial parameters are provided in Table 1, whereas the system's variables are defined in Table 2.

Let us derive the dynamic equations of the system. Suppose no sliding occurs between each wheel and the ground at the contact point, and that points A and B coincide ($d = 0$). Then, the kinematic relation between the yaw rate of the axle and the angular velocities of the wheels can be expressed as:

$$\dot{\theta} = \frac{r}{2a}(\dot{\theta}_r - \dot{\theta}_l) \tag{1}$$

The absolute forward speed v of point B (and A) of the axle can be expressed through the following nonholonomic relation:

$$v = \frac{r}{2}(\dot{\theta}_r + \dot{\theta}_l) \tag{2}$$

Projecting Eq. (2) along the unit vectors $\mathbf{i}_s$ and $\mathbf{j}_s$ of the fixed frame $\{S\}$, and differentiating with respect to time, we obtain the following expressions for the linear acceleration of point B of the axle in $\{S\}$:

$$\begin{cases} \ddot{x}_B = \frac{r}{2}\cos\theta(\ddot{\theta}_r + \ddot{\theta}_l) - \frac{r}{2}\sin\theta(\dot{\theta}_r + \dot{\theta}_l)\dot{\theta} \\ \ddot{y}_B = \frac{r}{2}\sin\theta(\ddot{\theta}_r + \ddot{\theta}_l) + \frac{r}{2}\cos\theta(\dot{\theta}_r + \dot{\theta}_l)\dot{\theta} \end{cases} \tag{3}$$

The equilibrium equations of the pendulum around the spherical joint in B read:

$$\mathfrak{J}_{p,B}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathfrak{J}_{p,B}\boldsymbol{\omega} = BG \times \mathbf{F}_B \tag{4}$$

where $\mathfrak{J}_{p,B}$ indicates the inertia tensor of the pendulum with respect to joint B and $\boldsymbol{\omega}$ its absolute angular velocity, both expressed in the body-fixed frame $\{B\}$. BG is the vector connecting point B to point G, where the center of mass of the pendulum is located, whereas $\mathbf{F}_B$ can be interpreted as a virtual force acting on the center of mass of the pendulum, caused by the acceleration of point B, and including also the effect of gravity. In the body-fixed reference frame $\{B\}$, $\mathbf{F}_B$ and BG can be expressed as:

$$\mathbf{F}_B = -m_p\mathbf{R}_{sb}^{\top}\begin{pmatrix} \ddot{x}_B \\ \ddot{y}_B \\ g \end{pmatrix} \qquad BG = \begin{pmatrix} 0 \\ 0 \\ l \end{pmatrix} \tag{5}$$

In order to describe the orientation of the pendulum in space, a singularity-free parameterization of $SO(3)$ must be employed. In this paper, two non-singular parameterizations of the rotation matrix $\mathbf{R}_{sb}$ are considered: one is based on the unit quaternion, the other on the DCM representation. In the following, the kinematic equations for both representations are presented.

2.1 Quaternion parameterization

Quaternions are a number system that extends complex numbers to a four-dimensional space. A quaternion $\mathbf{q}$ may be indicated as:

$$\mathbf{q} = (q_0, \mathbf{q}_v) \tag{6}$$

$\mathbf{q}$ consists of a scalar part q_0 and a three-dimensional vector part $\mathbf{q}_v$, which can be written as $\mathbf{q}_v = q_i\mathbf{e}_i$, where $\mathbf{e}_i$ represents each direction of the standard orthonormal basis in a 3D space for $i \in \{1,2,3\}$. The quaternion's conjugate $\mathbf{q}^*$ is defined as:

$$\mathbf{q}^* = (q_0, -\mathbf{q}_v) \tag{7}$$

Let us define the product of two quaternions $\mathbf{q}$ and $\mathbf{p}$ as follows:

$$\mathbf{q}\cdot\mathbf{p} = (q_0p_0 - \mathbf{q}_v^{\top}\mathbf{p}_v, q_0\mathbf{p}_v + p_0\mathbf{q}_v + \mathbf{q}_v \times \mathbf{p}_v) \tag{8}$$

The square root of the product of a quaternion and its conjugate is called quaternion norm, indicated here as $\|\mathbf{q}\|$:

$$\|\mathbf{q}\|^2 = \mathbf{q}\cdot\mathbf{q}^* = (q_0^2 + \mathbf{q}_v^{\top}\mathbf{q}_v, \mathbf{0}) \tag{9}$$

Unit quaternions belong to the group $SU(2)$ and therefore satisfy the unit length condition:

$$\|\mathbf{q}\| - 1 = \sqrt{q_0^2 + \mathbf{q}_v^{\top}\mathbf{q}_v} - 1 = 0 \tag{10}$$

A unit quaternion can parameterize the rotation matrix $\mathbf{R}_{sb}$ in every point of $SO(3)$ according to the following relation [28]:

$$\mathbf{R}_{sb}(\mathbf{q}) = \frac{1}{4}\mathbf{G}_s(\mathbf{q})\mathbf{G}_b(\mathbf{q})^{\top} \tag{11}$$

with

$$\mathbf{G}_s(\mathbf{q}) = 2\begin{bmatrix} -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{12}$$

$$\mathbf{G}_b(\mathbf{q}) = 2\begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \tag{13}$$

The quaternion dynamics can be expressed as:

$$\dot{\mathbf{q}} = \frac{1}{4}\mathbf{G}_b(\mathbf{q})^{\top}\boldsymbol{\omega} \tag{14}$$

Since $\mathbf{G}_b(\mathbf{q})\mathbf{q} = 0$ holds, it is straightforward to prove that Eq. (14) preserves the norm $\|\mathbf{q}\|$ of the quaternion over time.

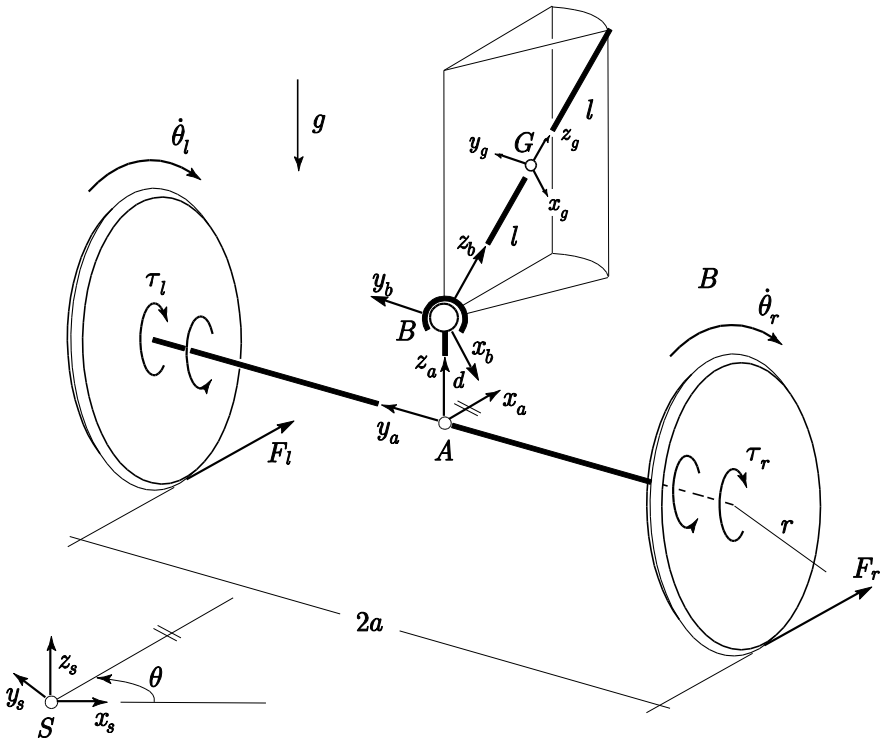


Fig. 1 Schematic of the benchmark system employed in the optimal control problem.

Table 1 Geometric and inertial parameters.

Parameter	Definition	Value
r	Wheel radius	0.05 m
a	Semi-length of the axle	0.10 m
l	Semi-length of the pendulum	0.25 m
d	Distance between point A in the middle of the axle and the spherical joint B	0
g	Acceleration of gravity	$9.81\,\mathrm{m\,s^{-2}}$
m_w	Mass of the wheel	0.25 kg
J_w	Moment of inertia of the wheel wrt its axis	$3.25 \cdot 10^{-4}\,\mathrm{kg\,m^2}$
J_w^d	Moment of inertia of the wheel wrt its diameter	$1.625 \cdot 10^{-4}\,\mathrm{kg\,m^2}$
m_a	Mass of the axle	0.5 kg
J_a	Moment of inertia of the axle wrt axis z_a	$1.66 \cdot 10^{-3}\,\mathrm{kg\,m^2}$
m_p	Mass of the pendulum	1.0 kg
$\mathfrak{I}_{p,G}$	Inertia tensor of the pendulum wrt $\{G\}$ reference frame	$J_{p,Gx} = J_{p,Gy} = 10J_{p,Gz} = 2.08 \cdot 10^{-2}\,\mathrm{kg\,m^2}$

2.2 DCM parameterization

Parameterizing the rotation $\mathbf{R}_{sb}$ via a DCM description consists in describing such rotation by using all of the elements of $\mathbf{R}_{sb}$ as parameters. Such elements represent the direction cosines of each axis of the body-fixed reference frame $\{B\}$ with respect to the fixed reference frame $\{S\}$, see [29]. In order for a matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ to be a valid DCM, it must belong to $SO(3)$, i.e. it must satisfy the orthonormality condition:

$$\mathbf{R}^\top \mathbf{R} - \mathbf{I} = 0 \tag{15}$$

Let us introduce the operator $\hat{\cdot} : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$ that maps any three-dimensional vector $\mathbf{v}$ into its skew-symmetric matrix form $\hat{\mathbf{v}}$, defined as:

$$\hat{\mathbf{v}} = \begin{pmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{pmatrix} \tag{16}$$

(v_x, v_y, v_z) being the components of the representation of $\mathbf{v}$ in a generic frame of reference.

It can be easily verified that orthonormality is preserved over time through the dynamics of the DCM, which reads:

$$\dot{\mathbf{R}} = \mathbf{R}\hat{\omega} \tag{17}$$

Here, $\hat{\omega} \in so(3)$ indicates the skew-symmetric matrix form of the angular velocity ω, expressed in the body-fixed reference frame.

Table 2 Definition of the system's variables.

Symbol	Definition
θ_r	Rotation angle of the right wheel
θ_l	Rotation angle of the left wheel
θ	Yaw angle of the axle
x_B	x_s coordinate of point B (and A)
y_B	y_s coordinate of point B (and A)
$\mathbf{a}_B$	absolute acceleration of point B (and A)
x_G	x_s coordinate of the center of mass G of the pendulum
y_G	y_s coordinate of the center of mass G of the pendulum
z_G	z_s coordinate of the center of mass G of the pendulum
$\mathbf{a}_G$	absolute acceleration of the center of mass G of the pendulum
F_r	Longitudinal component of the friction force acting on the right wheel
F_l	Longitudinal component of the friction force acting on the left wheel
τ_r	Internal torque acting on the right wheel
τ_l	Internal torque acting on the left wheel
$\mathbf{R}_{sb}$	Rotation matrix describing the orientation of the body-fixed frame $\{B\}$ wrt the fixed frame $\{S\}$
v	Norm of the absolute velocity of point B (and A) of the axle
ω	Body-fixed frame $\{B\}$ components of the absolute angular velocity of the pendulum

2.3 Equations of Motion

Let us briefly summarize the Equations of Motion (EOM) expressing the dynamics of the system. We consider two different formalizations of the same problem, differing only in the parameterization of the 3D orientation of the spherical pendulum. They are parameterized respectively by:

1. unit quaternion (Q)
2. DCM

2.3.1 Case 1: Unit quaternion (Q)

In case 1, the state vector $\mathbf{z}$ of the system consists of:

$$\mathbf{z} = (\mathbf{q}, x_B, y_B, \theta, \omega, v)^\top \tag{18}$$

For future reference, we conveniently partition the global state into two subvectors, respectively defined as:

$$\mathbf{q}, \qquad \chi = (x_B, y_B, \theta, \omega, v)^\top \tag{19}$$

Given this state-space description, the EOM of the system can be expressed as follows:

$$\dot{\mathbf{q}} = f_q(\mathbf{q}, \chi) = \frac{1}{4}\mathbf{G}_b(\mathbf{q})^\top \omega \tag{20}$$

$$\dot{\chi} = f_\chi(\mathbf{q}, \chi, \mathbf{u}) \tag{21}$$

The controls are chosen as

$$\mathbf{u} = \begin{pmatrix} u_\theta \\ u_v \end{pmatrix} = \begin{pmatrix} \dot{\theta} \\ \dot{v} \end{pmatrix}. \tag{22}$$

This implies full authority on the 2D kinematics of the cart and allows to decouple the dynamics of the wheel/cart subsystem from that of the pendulum, which enhances the numerical tractability of the system. Were we interested in computing the corresponding torques (τ_r, τ_l) acting on the wheels, we would use the given controls (u_θ, u_v), the computed trajectory $\mathbf{z}$ and its time derivative $\dot{\mathbf{z}}$, along with Eqs. (1), (2) and their time derivatives, to compute, via inverse dynamics, the longitudinal forces (F_r, F_l) and then (τ_r, τ_l). The procedure would trace the following steps. From the translational equilibrium of the whole system along $\mathbf{i}_a$, Eq. (23), and the rotational equilibrium of the wheel/cart subsystem along $\mathbf{k}_s$ through point B, Eq. (24), one would obtain F_r and F_l by solving the following system

$$F_r + F_l = (m_a + 2m_w)\dot{v} + m_p \mathbf{a}_G \cdot \mathbf{i}_a \tag{23}$$

$$(F_r - F_l)a = (J_a + 2J_w^d + 2m_w a^2)\ddot{\theta} \tag{24}$$

where $\mathbf{a}_G$ is readily computed as follows

$$\mathbf{a}_G = \mathbf{a}_B + \mathbf{R}_{sb}(\hat{\dot{\omega}} + \hat{\omega}^2)BG \tag{25}$$

At this point, the dynamics of the right and left wheels around their respective axes, which reads:

$$\begin{cases} J_w \ddot{\theta}_r = \tau_r - F_r r \\ J_w \ddot{\theta}_l = \tau_l - F_l r \end{cases}, \tag{26}$$

would provide the sought input torques τ_r and τ_l. It is worth mentioning that, despite limits on the kinematic controls u_θ and u_v are taken into account to model actuator saturation (as shall be stated in Eq. (32)), this does not ensure that possible limits on torques are met, as this verification would be done only *a posteriori*. A suitable way to circumvent this problem is setting the limits on $\mathbf{u}$ as conservatively as possible.

However, given the current choice of states and controls in Eqs. (19) and (22), the dynamics f_χ reads:

$$\dot{\chi} = f_\chi(\mathbf{q}, \chi, \mathbf{u}) = \begin{pmatrix} u_v \cos\theta - v u_\theta \sin\theta \\ u_v \sin\theta + v u_\theta \cos\theta \\ u_\theta \\ \mathfrak{J}_{p,B}^{-1}\left[\widehat{BG}\mathbf{F}_B - \hat{\omega}(\mathfrak{J}_{p,B}\omega)\right] \\ u_v \end{pmatrix} \tag{27}$$

2.3.2 Case 2: Direction Cosine Matrix (DCM)

In case 2, the state $\mathbf{z}$ is parameterized as:

$$\mathbf{z} = (\mathbf{R}, x_B, y_B, \theta, \omega, v)^\top \tag{28}$$

which can be partitioned into:

$$\mathbf{R}, \qquad \chi = (x_B, y_B, \theta, \omega, v)^\top \tag{29}$$

Taking into account all the previous considerations together with Eq. (17), the EOM of the system are given in this case by:

$$\dot{\mathbf{R}} = f_R(\mathbf{R}, \chi) = \mathbf{R}\hat{\omega} \tag{30}$$

$$\dot{\chi} = f_\chi(\mathbf{R}, \chi, \mathbf{u}) \tag{31}$$

In the following, the dynamics of the system will be referred to as $\dot{\mathbf{z}} = f(\mathbf{z}, \mathbf{u})$, i.e. Eqs. (20) and (21) together in case 1, or Eqs. (30) and (31) together in case 2.

3 Trajectory optimization

In this section, the trajectory optimization problem is presented. Then, the main issues related to the presence of invariants in the dynamics, due to non-minimal parameterizations of $SO(3)$, are addressed. Finally, the numerical method employed to solve the problem is introduced.

3.1 Optimal Control Problem formulation

The Optimal Control Problem (OCP) whose solution contains a sequence of control actions that steer the unicycle from a given initial state to a predefined one, within a given time horizon T, is expressed by:

$$\begin{aligned} \min_{\mathbf{z}(\cdot),\mathbf{u}(\cdot)} \quad & \int_0^T L(\mathbf{z}(t), \mathbf{u}(t))dt \\ \text{subject to} \quad & \mathbf{z}(0) = \mathbf{z}_{\text{in}} \\ & \mathbf{z}(T) = \mathbf{z}_{\text{des}} \\ & \dot{\mathbf{z}}(t) = f(\mathbf{z}(t), \mathbf{u}(t)), \quad t \in [0, T] \\ & h(\mathbf{u}(t)) \leq 0, \qquad\quad t \in [0, T] \end{aligned} \tag{32}$$

Here, the symbols $\mathbf{z}$ and $\mathbf{u}$ denote, respectively, states and controls of the system. In the inequality constraints $h(\cdot)$, limits on the controls are imposed, so as to model actuator saturation.

Among all the possible trajectories that bring the system from the initial state $\mathbf{z}_{\text{in}}$ to the desired final state $\mathbf{z}_{\text{des}}$ while satisfying the dynamics, we seek one which minimizes an integral cost function. Specifically, the minimized objective function consists of three terms, respectively aimed at:

- minimizing the difference between the current and the desired position of the axle,
- maximizing the gravitational potential energy, so as to penalize the deviation of the current orientation of the pendulum from the desired (vertical) one,
- minimizing the time derivatives of the controls, so as to smooth the optimal trajectory.

Therefore, the Lagrange-type objective function $L(\cdot)$ is designed as

$$L(\mathbf{z}, \mathbf{u}) = k_1 \Delta(\mathbf{z}) - k_2 m_p z_G g + k_3 \left\| \frac{d\mathbf{u}}{dt} \right\|^2 \tag{33}$$

where the weighting coefficients $(k_1, k_2, k_3) > 0$ are chosen once for all, in such a way that the three terms are appropriately scaled and thus their contributions to the cost function have approximately the same order of magnitude in absolute value. The function $\Delta(\mathbf{z})$ aims to minimize the difference between the current and the final position of the axle, and it is defined as:

$$\Delta(\mathbf{z}) = c_x \|x_B - x_{\text{des}}\|^2 + c_y \|y_B - y_{\text{des}}\|^2 + c_\theta \|\theta - \theta_{\text{des}}\|^2 \tag{34}$$

where $(x_{\text{des}}, y_{\text{des}})$ and θ_{des} respectively identify the desired final Cartesian position and orientation of the axle, and the coefficients (c_x, c_y, c_θ) were selected to appropriately scale the different measures.

3.2 Optimal control for equations with invariants

We aim to optimize a motion between two specified states $\mathbf{z}_{\text{in}}$ and $\mathbf{z}_{\text{des}}$. Since in both formalizations (cases 1 and 2) the parameterization of the rotational DOF of the pendulum is non-minimal, the components of the state vector which parameterize $SO(3)$ belong to a nonlinear manifold. States $\mathbf{z}_{\text{in}}$ and $\mathbf{z}_{\text{des}}$ identify two points belonging to the same manifold. Moreover, the equations describing the dynamics of the $SO(3)$ parameterization, namely f_q in case 1 and f_R in case 2, respectively preserve the norm of the quaternion and the orthonormality of the DCM. In fact, as already mentioned in Section 2, these quantities are invariant in the considered dynamics. As a result, the constraints of the nominal OCP:

$$\begin{cases} \mathbf{z}(0) = \mathbf{z}_{\text{in}} \\ \mathbf{z}(T) = \mathbf{z}_{\text{des}} \\ \dot{\mathbf{z}}(t) = f(\mathbf{z}(t), \mathbf{u}(t)) \end{cases} \tag{35}$$

turn out to be redundant in both formulations. Let us indicate a generic invariant by the notation $c(\mathbf{z}) = 0$. Recall that, in our case, this condition identifies the manifold the considered parameterization of $SO(3)$ belongs to. Since the dynamics f intrinsically preserves this invariant over time [30], the condition $c(\mathbf{z}) = 0$ needs to be imposed only at one point of the trajectory $\mathbf{z}(t)$, and this condition will be sufficient to

guarantee that the whole trajectory $\mathbf{z}(t)$ evolves on the manifold. Therefore, imposing both conditions

$$\begin{cases} c(\mathbf{z}_{\text{in}}) = 0 \\ c(\mathbf{z}_{\text{des}}) = 0 \end{cases} \tag{36}$$

is redundant, and the constraint equations (35) are linearly dependent: this implies that the Linear Independence Constraint Qualification (LICQ) condition does not hold [31], causing problems in the numerical solution.

To prevent the OCP from violating the LICQ, a null-space approach, as proposed in [32], can be applied. Namely, this approach consists in substituting the nominal constraints (35) of the OCP with an equivalent set of constraints, where the redundant condition is eliminated by projection into the null-space of the Jacobian $\frac{\partial c}{\partial \mathbf{z}}$. Consequently, the equivalent OCP reads:

$$\begin{aligned} \min_{\mathbf{z}(\cdot),\mathbf{u}(\cdot)} \quad & \int_0^T L(\mathbf{z}(t),\mathbf{u}(t))dt \\ \text{subject to} \quad & \mathbf{z}(0) = \mathbf{z}_{\text{in}} \\ & \mathbf{W}^\top(\mathbf{z}(T) - \mathbf{z}_{\text{des}}) = \mathbf{0} \\ & \dot{\mathbf{z}}(t) = f(\mathbf{z}(t),\mathbf{u}(t)), \qquad t \in [0,T] \\ & h(\mathbf{u}(t)) \leq 0, \qquad t \in [0,T] \end{aligned} \tag{37}$$

where $\mathbf{W}$ is an orthonormal basis of the null-space of the Jacobian matrix $\mathbf{J} = \frac{\partial c}{\partial \mathbf{z}}(\mathbf{z}(T))$, which is assumed to have full rank.

3.3 Numerical solution

There are several ways to obtain the numerical solution of a trajectory optimization problem [33]. In this work a direct multiple shooting method [34] is applied to solve the OCP (37). This method consists in parameterizing the continuous time OCP into a finite-dimensional nonlinear program (NLP). The time horizon T is discretized into N time intervals $[t_k, t_{k+1}]$, such that $t_0 = 0$, $t_N = T$ and $k \in \{0,1,\ldots,N-1\}$. Accordingly, the trajectories of the states $\mathbf{z}(t)$ and controls $\mathbf{u}(t)$ are parameterized respectively by the sequences $\{\mathbf{z}_k\}$ and $\{\mathbf{u}_k\}$ of optimization variables. The controls $\mathbf{u}_k$ are assumed to be piecewise constant over each time interval $[t_k, t_{k+1}]$, whereas, in order to relate the current state $\mathbf{z}_k$ to the value $\mathbf{z}_{k+1}$ it assumes at the end of the subsequent time interval, the Ordinary Differential Equation (ODE) describing the dynamics of the system is integrated numerically over the period $[t_k, t_{k+1}]$, starting from the initial value $\mathbf{z}_k$:

$$\begin{cases} \dot{\mathbf{z}}(t;\mathbf{z}_k,\mathbf{u}_k) = f(\mathbf{z}(t;\mathbf{z}_k,\mathbf{u}_k),\mathbf{u}_k) \\ \mathbf{z}(t_k;\mathbf{z}_k,\mathbf{u}_k) = \mathbf{z}_k \end{cases} \tag{38}$$

The integral cost function is numerically approximated consistently.

Thus, the original continuous time OCP results in the following finite-dimensional NLP:

$$\begin{aligned} \min_{\{\mathbf{z}_k\},\{\mathbf{u}_k\}} \quad & \sum_{k=0}^{N-1} L(\mathbf{z}_k,\mathbf{u}_k)(t_{k+1} - t_k) \\ \text{subject to} \quad & \mathbf{z}_0 = \mathbf{z}_{\text{in}} \\ & \mathbf{W}^\top(\mathbf{z}_N - \mathbf{z}_{\text{des}}) = \mathbf{0} \\ & \mathbf{z}_{k+1} - \mathbf{z}(t_{k+1};\mathbf{z}_k,\mathbf{u}_k) = \mathbf{0}, \quad \forall k \in \{0,N-1\} \\ & h(\mathbf{u}_k) \leq 0, \qquad \forall k \in \{0,N-1\} \end{aligned} \tag{39}$$

This NLP is equivalently rewritten in the following form:

$$\begin{aligned} \min_{\nu} \quad & F(\nu) \\ \text{subject to} \quad & g_i(\nu) = 0, \qquad \forall i \in \mathscr{E} \\ & g_i(\nu) \leq 0, \qquad \forall i \in \mathscr{I} \end{aligned} \tag{40}$$

in which ν indicates a vector collecting all the decision variables of the NLP (39). The vector function $g(\cdot)$ describes all the constraints, $\mathscr{E}$ and $\mathscr{I}$ being the sets of indices corresponding, respectively, to equalities and inequalities. Their cardinalities are denoted, respectively, by the symbols m_e and m_i. Any local minimizer ν^* of problem (40) must satisfy the first-order necessary conditions of optimality. These conditions involve the stationarity of the Lagrangian, defined as follows:

$$\mathscr{L}(\nu,\lambda) = F(\nu) + \lambda^\top g(\nu)$$

where $\lambda \in \mathbb{R}^{m_i+m_e}$ is a vector collecting the *Lagrange multipliers*, or dual variables. If ν^* is a local minimum satisfying LICQ, then a unique vector λ^* exists such that the following relations, often referred to as Karush-Kuhn-Tucker (KKT) conditions [31], hold:

$$\nabla F(\nu^*) + \sum_{i \in \mathscr{I} \cup \mathscr{E}} \lambda_i^* \nabla g_i(\nu^*) = \mathbf{0}, \tag{41}$$

$$g_i(\nu^*) = 0, \qquad \forall i \in \mathscr{E} \tag{42}$$

$$g_i(\nu^*) \leq 0, \qquad \forall i \in \mathscr{I} \tag{43}$$

$$\lambda_i^* g_i(\nu^*) = 0, \quad \lambda_i^* \geq 0, \qquad \forall i \in \mathscr{I} \tag{44}$$

Specifically, in Eq. (41) stationarity of the Lagrangian at (ν^*,λ^*) is required, Eqs. (42) and (43) imply that ν^* must belong to the feasible set, and in Eq. (44) a complementarity condition on the dual variables corresponding to inequality constraints is stated. The KKT conditions form a system of nonlinear equations, which can be solved numerically via Newton-like methods [18]. In this paper, a primal-dual interior-point solver is employed in order to compute local minimizers (ν^*,λ^*) of (39). Thus, the obtained optimal trajectory $\{\mathbf{z}_k^*\}$ approximates a continuous-time state trajectory $\mathbf{z}^*(t)$ that locally minimizes (37) with an accuracy that depends on the accuracy of the integration scheme used for the integration of (38). This approximation is consistent with that to which the control sequence $\{\mathbf{u}_k^*\}$ approximates

the corresponding optimal control trajectory $\mathbf{u}^*(t)$. Among the Lagrange multipliers λ^*, those relative to the continuity constraints $\mathbf{z}_{k+1} - \mathbf{z}(t_{k+1};\mathbf{z}_k,\mathbf{u}_k) = \mathbf{0}$ constitute an approximation of the costates of the OCP (37), see e.g. [35]. In [36], it was shown that this approximation has a lower order of accuracy. It is worth emphasizing that, in the event that the same order of accuracy were pursued for both the primal and the dual variables, specific direct methods might be employed that, under certain assumptions [37], can avoid this inconvenience.

4 Integration schemes

The need for numerically integrating the dynamics f of the system over each time interval $[t_k, t_{k+1}]$, i.e. solving the initial value problem (IVP) stated in (38), poses some issues when f intrinsically preserves some invariant quantity. In fact, in order to avoid redundancy in the constraints of the OCP, the condition $c(\mathbf{z}) = 0$, which in our case ensures that the non-minimal parameterization of a rotation belongs to the correct nonlinear manifold, must be imposed just at one point of the state trajectory, which was chosen as $\mathbf{z}_0$ in (39). When the integration is carried out using standard integrators for long simulation horizons, numerical errors cause the state trajectory to drift away from the underlying $SO(3)$ manifold, which makes the computed solution inaccurate. However, specific geometric integration techniques exist which guarantee that the solution of the IVP preserves the structural properties of the Lie group to numerical accuracy.

In this section, standard numerical integration schemes are briefly recalled, mostly for setting the notation. Furthermore, the most popular techniques used to handle the drift due to numerical integration errors are addressed. Afterwards, two structure-preserving numerical integrators, respectively acting on the Special Unitary group $SU(2)$ and on the Special Orthogonal group $SO(3)$ are introduced. The efficiency of the application of these integration schemes for the solution of our benchmark trajectory optimization problem will be analyzed in Section 5.

4.1 Standard RK integrators

Let us briefly describe the classical explicit RK integration method [38]. Suppose that a differential equation $\dot{\mathbf{y}}(t) = f(\mathbf{y}(t),t)$ is given and that the state value $\mathbf{y}_k$ at time $t = t_k$ is known. Then, a numerical method can be employed in the solution of the IVP

$$\begin{cases} \dot{\mathbf{y}}(t) = f(\mathbf{y}(t),t) \\ \mathbf{y}(t_k) = \mathbf{y}_k \end{cases} \tag{45}$$

Specifically, the s-stage explicit RK method consists in approximating the solution $\mathbf{y}_{k+1}$ of the differential equation at time $t_{k+1} = t_k + h$ by applying the following algorithm:

$\mathbf{y}^{(i)} \leftarrow \mathbf{y}_k$
for $i = 1,2,\ldots,s$ **do**
  $\boldsymbol{\kappa}^{(i)} \leftarrow f(\mathbf{y}^{(i)}, t_k + c_i h)$
  $\mathbf{y}^{(i)} \leftarrow \mathbf{y}_k + h\sum_{j=1}^{i-1} a_{ij}\boldsymbol{\kappa}^{(j)}$
end for
$\mathbf{y}_{k+1} = \mathbf{y}_k + h\sum_{i=1}^{s} b_i \boldsymbol{\kappa}^{(i)}$

where the coefficients c_i, a_{ij}, b_i are defined by the respective Butcher table.

As already mentioned, such integration schemes are not tailored to be employed in the integration of quantities evolving on a nonlinear manifold. In fact, in the event that $\mathbf{y}(t)$ belongs to a Lie group, despite the structure of the group is invariant through the dynamics $\dot{\mathbf{y}}(t) = f(\mathbf{y}(t),t)$, a classical RK integrator does not guarantee its preservation.

In order to tackle this problem, stabilization techniques [39] aimed at reducing the drift from the manifold due to numerical integration errors may be employed. Among them, the most popular is Baumgarte's method, which consists in adding a fictitious dynamics to the differential equations evolving on a manifold. This correction term ensures the damping of numerical integration errors, such that the solution of the differential equation converges to the manifold. The application of this method to the stabilization of the dynamics of a unit quaternion or a DCM, i.e. Eqs. (20) and (30), results respectively in:

$$\dot{\mathbf{q}} = \frac{1}{4}\mathbf{G}_b(\mathbf{q})^\top \boldsymbol{\omega} + \gamma_q \left((\|\mathbf{q}\|^2)^{-1} - 1 \right) \mathbf{q} \tag{46}$$

$$\dot{\mathbf{R}} = \mathbf{R}\hat{\boldsymbol{\omega}} + \gamma_R(\mathbf{R}^\top\mathbf{R} - \mathbf{I})\mathbf{R} \tag{47}$$

where the coefficients γ_q and γ_R must be calibrated so as to guarantee the stability of the above differential equations. This calibration should consider the dynamics of the drift, which should be made as fast as the numerical issues related to the stiffness of the resulting equations allow. In the analysis presented in the following, Baumgarte's coefficients were calibrated by performing extensive numerical tests aimed at assessing the sensitivity of the stabilized dynamics to their variation, in accordance with the guidelines in [7]. As a result, $\gamma_q = 2$ and $\gamma_R = 1$ were chosen.

It is worth mentioning that Baumgarte's stabilization has been successfully applied to the stabilization of singularity-free parameterizations of $SO(3)$ in the context of direct optimal control in the recent contributions [40, 41].

4.2 Quaternion integrator

As a redundant parameterization, also quaternion integration must be handled with care. In fact, since the unit length condition stated in Eq. (10) must be satisfied in order for the

quaternion to belong to $SU(2)$, the integration of a Differential Algebraic Equation (DAE) has to be addressed. To circumvent this problem, Lie group methods, such as Crouch Grossmann or Munthe-Kaas, were recently applied to the integration of quaternions [17, 42]. In this paper we refer to the integrator proposed in [17], which is briefly reviewed here. For a more detailed explanation, the reader is referred to [17] and the references therein.

Let the kinematic reconstruction equation of the considered rotation $\mathbf{R}_{sb}$ be:

$$\dot{\mathbf{R}}_{sb}(t) = \mathbf{R}_{sb}(t)\hat{\omega} \tag{48}$$

where ω is the angular velocity expressed in the body-fixed reference frame and $\hat{\omega} \in so(3)$ its skew-symmetric matrix form. This ODE system evolves on the Lie group $SO(3)$, thus the solution is formally given by:

$$\mathbf{R}_{sb}(t) = \mathbf{R}_{sb}(t_0)\exp_{SO(3)}(\hat{\mathbf{w}}) \tag{49}$$

where $\hat{\mathbf{w}} \in so(3)$ is the instantaneous rotation vector and the exponential mapping is defined in closed form by the Rodrigues' formula [43]:

$$\exp_{SO(3)}(\hat{\mathbf{w}}) = \mathbf{I} + \frac{\sin\|\mathbf{w}\|}{\|\mathbf{w}\|}\hat{\mathbf{w}} + \frac{1-\cos(\|\mathbf{w}\|)}{\|\mathbf{w}\|}\hat{\mathbf{w}}^2 \tag{50}$$

The instantaneous rotation vector $\mathbf{w}$ solves the following ODE system [9, 15]:

$$\dot{\mathbf{w}} = \mathrm{dexp}^{-1}_{\mathbf{w}}\,\omega \tag{51}$$

with the initial value $\mathbf{w}(t_0) = \mathbf{0}$.
The operator $\mathrm{dexp}^{-1}_{\mathbf{w}} : so(3) \times so(3) \to so(3)$ is given in closed form by the formula:

$$\mathrm{dexp}^{-1}_{\mathbf{w}} = \mathbf{I} + \frac{1}{2}\hat{\mathbf{w}} - \frac{\|\mathbf{w}\|\cot\left(\frac{\|\mathbf{w}\|}{2}\right) - 2}{2\|\mathbf{w}\|^2}\hat{\mathbf{w}}^2 \tag{52}$$

It is worth noticing that Eq. (51) is a differential equation evolving on the Lie algebra $so(3)$, which is a linear space. Therefore, in order to compute a solution for Eq. (14), the Munthe-Kaas (MK) method prescribes to first integrate this equation using a standard RK integrator, so as to obtain an instantaneous rotation vector $\mathbf{w}$. This vector is then mapped onto the group $SU(2)$ by means of the exponential mapping $\exp_{SU(2)}$, which relates elements on the tangential space to elements of the group $SU(2)$ itself. This map is also available in closed form:

$$\exp_{SU(2)}(\mathbf{w}) = \cos\left(\frac{1}{2}\|\mathbf{w}\|\right)(1,\mathbf{0}) + \frac{\sin\left(\frac{1}{2}\|\mathbf{w}\|\right)}{\|\mathbf{w}\|}(0,\mathbf{w}) \tag{53}$$

The computed quantity can be used subsequently to update the quaternion parameterization.

The implemented algorithm for the integration in the time interval $[t_k, t_k + h]$ is presented below:

$\mathbf{q}^{(i)} \leftarrow \mathbf{q}_k$, $\chi^{(i)} \leftarrow \chi_k$
for $i = 1, 2, \ldots, s$ **do**
  $\kappa_q^{(i)} \leftarrow f_q(\mathbf{q}^{(i)}, \chi^{(i)})$
  $\kappa_\chi^{(i)} \leftarrow f_\chi(\mathbf{q}^{(i)}, \chi^{(i)}, u_k)$
  $\chi^{(i)} \leftarrow \chi_k + h\sum_{j=1}^{i-1} a_{ij}\kappa_\chi^{(j)}$
  $\mathbf{w}^{(i)} \leftarrow h\sum_{j=1}^{i-1} a_{ij}\kappa_q^{(j)}$
end for
$\chi_{k+1} = \chi_k + h\sum_{i=1}^{s} b_i\chi^{(i)}$
$\mathbf{q}_{k+1} = \mathbf{q}_k \cdot \exp_{SU(2)}(h\sum_{i=1}^{s} b_i\mathbf{w}^{(i)})$

4.3 DCM integrator

Also the DCM presents some difficulties in its time integration due to the redundancy of the parameterization. In this case, the condition to be satisfied for the DCM to belong to the Lie group $SO(3)$ is the orthonormality of the parameterization $\mathbf{R}$, stated in Eq. (15). The MK method may be applied to the integration of Eq. (17), which is an ODE directly evolving on the Lie group $SO(3)$.

The solution of the kinematic reconstruction equation is given by Eq. (49). In order to solve it, its differential form can be first integrated numerically, namely Eq. (51), which evolves on the Lie algebra $so(3)$. Afterwards, the instantaneous rotation vector $\mathbf{w}$ can be mapped on the group $SO(3)$ through the corresponding exponential mapping $\exp_{SO(3)}$.

The integration algorithm in the time interval $[t_k, t_k + h]$ can be therefore implemented as follows:

$\mathbf{R}^{(i)} \leftarrow \mathbf{R}_k$, $\chi^{(i)} \leftarrow \chi_k$
for $i = 1, 2, \ldots, s$ **do**
  $\kappa_R^{(i)} \leftarrow f_R(\mathbf{R}^{(i)}, \chi^{(i)})$
  $\kappa_\chi^{(i)} \leftarrow f_\chi(\mathbf{R}^{(i)}, \chi^{(i)}, u_k)$
  $\chi^{(i)} \leftarrow \chi_k + h\sum_{j=1}^{i-1} a_{ij}\kappa_\chi^{(j)}$
  $\mathbf{w}^{(i)} \leftarrow h\sum_{j=1}^{i-1} a_{ij}\kappa_R^{(j)}$
end for
$\chi_{k+1} = \chi_k + h\sum_{i=1}^{s} b_i\chi^{(i)}$
$\mathbf{R}_{k+1} = \mathbf{R}_k \exp_{SO(3)}(h\sum_{i=1}^{s} b_i\mathbf{w}^{(i)})$

5 Numerical results

In order to assess the performance of structure-preserving integrators in trajectory optimization problems via direct multiple shooting, we compare the solutions obtained for our benchmark problem using different schemes for integrating the rotational DOF of the pendulum over the shooting segments. The OCP described in Section 3 was formulated in the CasADi framework [44], and the resulting NLP was solved using the interior-point solver IPOPT [45].

For the comparison, three different integration schemes were employed in the integration of the EOM of each of the dynamic models described in Section 2.3:

Table 3 Number of nodes in the graph representation of each model (row), integration-scheme (column) pair.

Model	Standard RK4	Standard RK4 with Baumgarte's stabilization	Lie group integrator (integration + exponential mapping)
Quaternion	639	712	$903+49$
DCM	521	810	$900+112$

Table 4 Average computational time (in ms) per exact Newton iteration for each model (row), integration-scheme (column) pair.

Model	Standard RK4	Standard RK4 with Baumgarte's stabilization	Lie group integrator
Quaternion	67	68	102
DCM	91	113	148

(a) Standard one-step explicit 4^{th}-order RK integrator, without any stabilization for the singularity-free parameterization of $SO(3)$ (hereafter RK)
(b) Standard one-step explicit 4^{th}-order RK integrator, where the singularity-free parameterization of $SO(3)$ is stabilized through Baumgarte's method (hereafter RK+BG)
(c) One-step 4^{th}-order RK-MK Lie group integrator, illustrated in Sections 4.2 and 4.3, respectively, for the quaternion and the DCM parameterization (hereafter RK-MK).

5.1 Model complexity

The software tool CasADi, employed for formulating and solving each OCP, enables efficient calculation of the derivatives required by Newton-based optimization solvers through state-of-the-art methods for Algorithmic Differentiation (AD) [46]. In order to perform AD, CasADi creates symbolic expressions for the objective function and the constraints, which are represented in memory as computational graphs. Since the number of nodes in the graph is related to the symbolic complexity of the problem to be solved, a comparison of the number of nodes related to the representation of the considered integrators in both dynamic models is presented. As evident from the results in Table 3, in both formulations the use of structure-preserving integrators implies a considerable increase in the symbolic complexity of the constraints of the OCP. This results in a corresponding increase in computational time. As reported in Table 4, the characteristic CPU time, i.e. the average time required to perform an exact Newton iteration for each formulation, is in fact significantly higher for structure-preserving integrators. It is also worth noting that stabilizing a non-minimal representation of $SO(3)$ with Baumgarte's method implies a slightly higher complexity in the model (cf. Table 3, 1^{st} and 2^{nd} columns). This effect is particularly noticeable in the DCM parameterized model (cf. Table 3, 2^{nd} column, 2^{nd} row). As a result, the computational time required in the computation of each Newton iteration is higher than the case in which the dynamics of the parameterization is not stabilized. The average CPU time per iteration of each formulation of the same OCP is reported in Table 4. These results were obtained on a desktop PC with 3.20 GHz Intel(R) Core(TM) i7-3930K CPU and 32 GB of RAM.

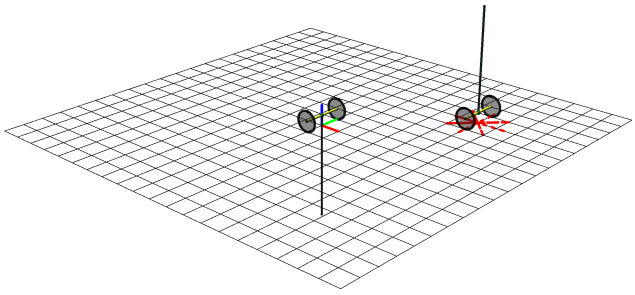
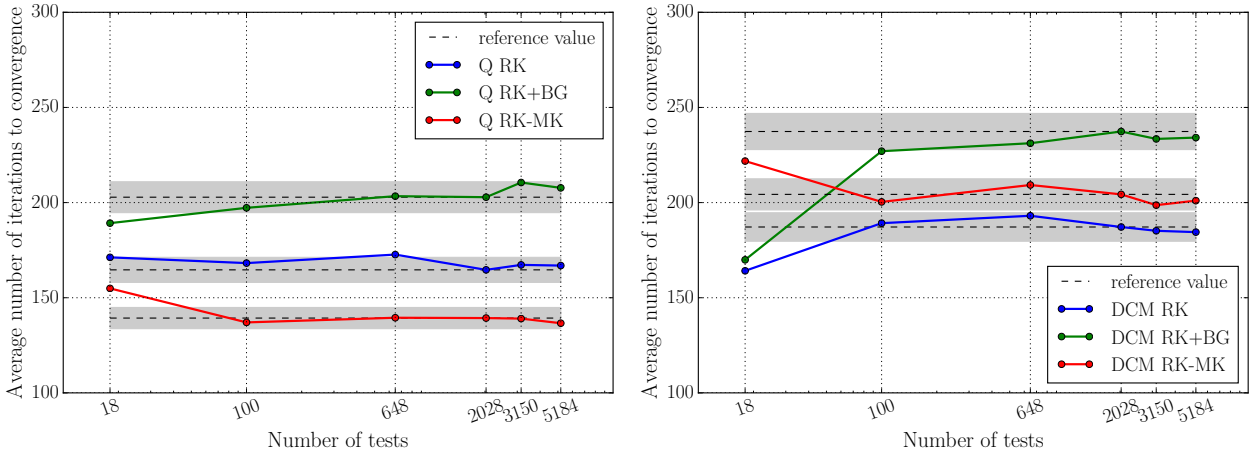


Fig. 2 Initial (pendulum downwards) and final (pendulum upwards) configurations prescribed to the system. Different scenarios were optimized: the final Cartesian position $(x_{\text{des}}, y_{\text{des}})$ of the axle was chosen on the uniform square grid having side 2 m. For each position, different final orientations θ_{des} were tested (see red dashed lines).

5.2 Task description

The performance of each formulation was evaluated in trajectory optimization of the nonholonomic spherical pendulum model described in Section 2. The task to be optimized consists in maneuvering the unicycle so as to steer the system from a given initial state, with the pendulum oriented downwards, to a target final state, with the pendulum in the upright position, within a time horizon of 3 seconds, see Fig. 2.

In order to gain insight into the effects of employing different parameterizations of $SO(3)$ and different types of integrators in direct optimal control, optimizations were performed on a (13x13)x12 lattice, for a total of 2028 different nodes (tasks), corresponding to a 13x13 grid of target Cartesian positions $(x_{\text{des}}, y_{\text{des}})$ in the interval $[-1,1]\times[-1,1]$ m,



(a) Q representation

(b) DCM representation

Fig. 3 Average number of iterations to convergence for different numbers of tests. For each integration scheme, the value corresponding to the selected number of tests to be performed (2028) is marked with a dashed line, and a confidence interval of $\pm 4\%$ around it is gray-shaded.

each one to be reached by the unicycle in 12 different yaw orientations θ_{des} in the interval $[0, 2\pi]$. The number of tests to be presented was chosen according to a convergence analysis, whose results in terms of average number of iterations to convergence are shown in Fig. 3. The same analysis was also carried out for the average minimum (optimal) value of the cost function. In fact, performing the tests on a two increasingly fine lattices with (15x15)x14=3150 points and (18x18)x16=5184 points, the output did not vary more than $\pm 4\%$ with respect to the reference lattice. The obtained solutions were compared in terms of value of the cost function at the optimum, drift of the non-singular parameterization of $SO(3)$ from the manifold, and computational burden required by the solver to converge.

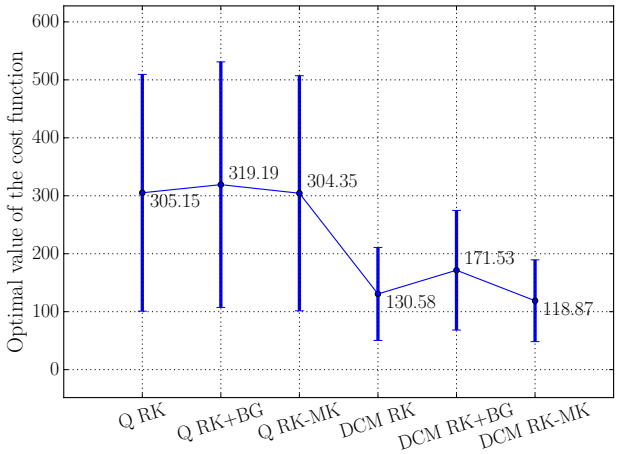


Fig. 4 Optimal value of the cost function: mean value $\pm$ standard deviation for each formulation.

5.3 Discussion

Let us provide some comments on the obtained results. In Fig. 4 the optimal values of the objective function for each formulation are presented. It can be observed that, in those cases where the solver found an optimal solution for the OCP, those obtained by using a DCM parameterization are, on average, characterized by a lower value of the objective function. This might be due to the fact that the EOM are more linear when a DCM representation is employed, compared to the one corresponding to a quaternion parameterization. It is worth pointing out that these results are consistent with those presented in [47], where different dynamic models of the same system, differing in the rotation parameterization, were compared: the author observed a better quality of the obtained optimal solutions when a DCM parameterization was employed. Interestingly, we can observe that the different minima of the cost function correspond to different local optima, each one encoding a different behaviour of the physical model over the optimized trajectory. Specifically, when the dynamic model is formulated using a quaternion parameterization, in the optimized trajectories the pendulum is usually swung up by initially accelerating the cart and then exploiting the centrifugal force produced by a sharp turn of the unicycle, as evident from Fig. 5. On the other hand, when a DCM representation is employed, the computed solution usually has a lower objective function value, corresponding to a trajectory in which the unicycle swings up the pendulum through heavy braking and then stabilizes it in the upright position via oscillatory motions, as in Fig. 6. Animations of frame sequences from the trajectories optimized for different tasks using different $SO(3)$ parameterizations can be found in the video [48] accompanying this pa-

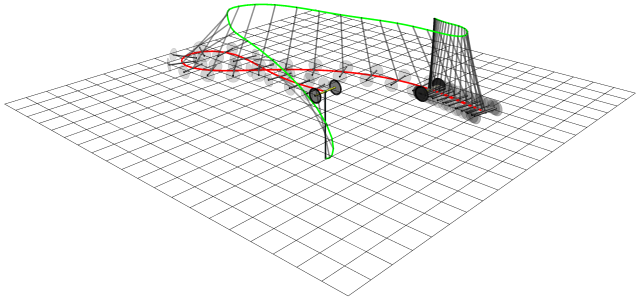


Fig. 5 Quaternion representation: time lapse of the optimal trajectory of the system with $x_{\text{des}} = 0.5\,\text{m}$, $y_{\text{des}} = 0.5\,\text{m}$, $\theta_{\text{des}} = \pi$. The red and the green lines represent, respectively, the trajectories of the spherical joint and of the tip of the pendulum.

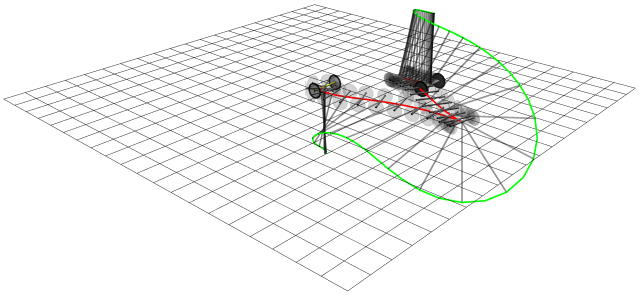
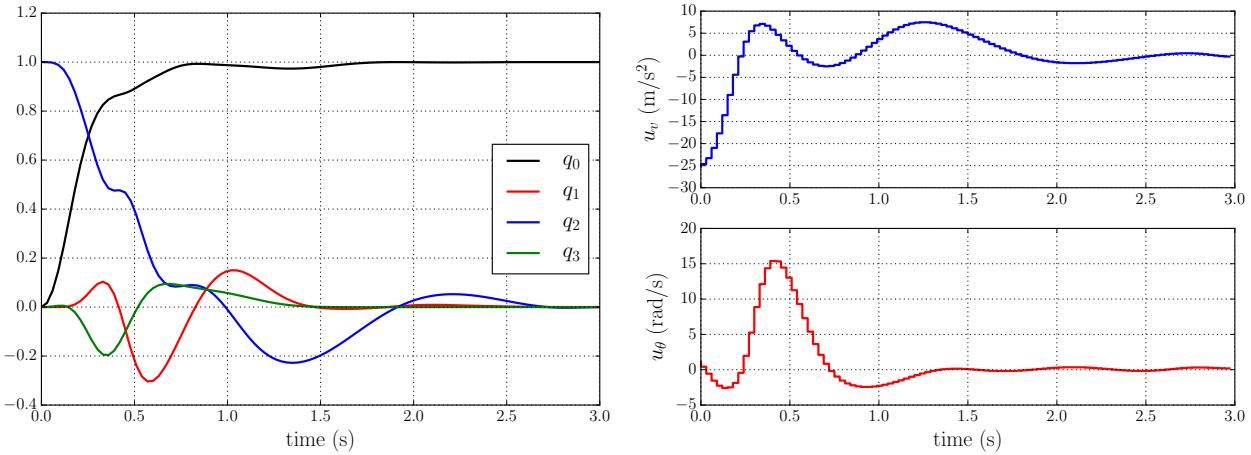


Fig. 6 DCM representation: time lapse of the optimal trajectory of the system with $x_{\text{des}} = 0.5\,\text{m}$, $y_{\text{des}} = 0.5\,\text{m}$, $\theta_{\text{des}} = \pi$. The red and the green lines represent, respectively, the trajectories of the spherical joint and of the tip of the pendulum.



(a) States **q** (b) Controls **u**

Fig. 7 Optimal trajectory obtained with $x_{\text{des}} = 0.5\,\text{m}$, $y_{\text{des}} = 0.5\,\text{m}$, $\theta_{\text{des}} = \pi$, when a quaternion representation is employed. (a), trajectory of quaternion components. (b), corresponding applied controls.

per. In addition, plots of a subset of the state trajectories and associated controls corresponding to the solution depicted in Fig. 5 are reported in Fig. 7.

In Fig. 8 the ratios of successfully solved OCPs over the total number of tests is presented. In both dynamic models, the use of Baumgarte's method and Lie group integrators have the effect of reducing the rate of success compared to a non-stabilized standard RK integration. However, success of the latter comes at a cost: the accuracy of non-stabilized integration schemes in terms of drift from $SO(3)$ is naturally lower than that obtained with structure-preserving integrators.

In Fig. 9, the amount of drift of solutions of the same OCP obtained by employing the different formulations is shown. The drift from the manifold might represent a problem in applications in which the time horizon to be considered is "rather long". Even in the analyzed case, where the time horizon is of 3 seconds only, the drift is perceivable, since it exceeds the accuracy of the integrator. In fact, a 4$^{\text{th}}$-order RK method is employed, which means the computed solution x_k coincides with the exact solution $x(t_k)$ up to $O(h^p)$, where p is the order of the integration method.

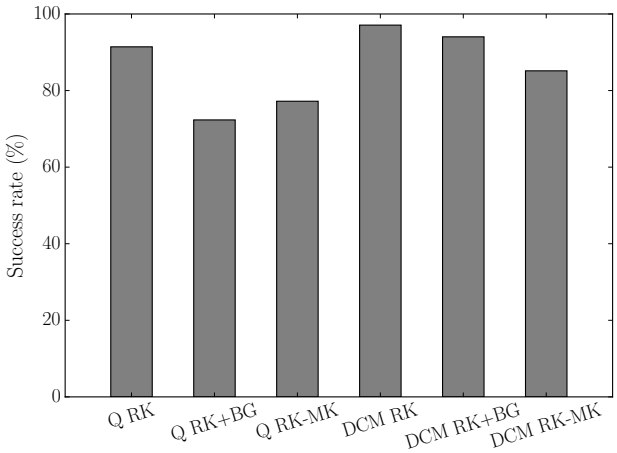
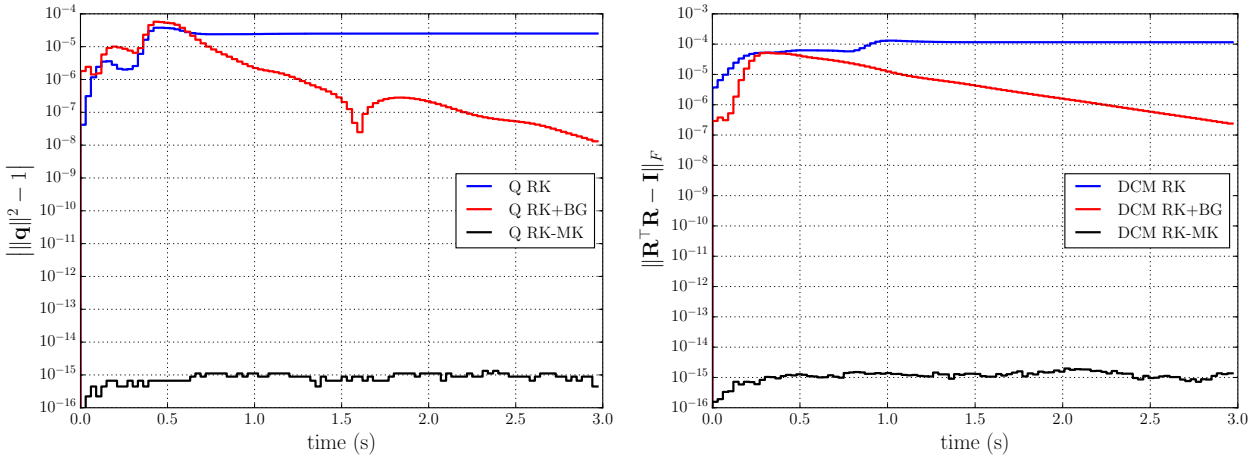


Fig. 8 Rate of successfully solved OCPs for each formulation.



(a) Q (b) DCM

Fig. 9 Deviation of each representation from the corresponding manifold. (a), deviation of the quaternion (Q) representation from $SU(2)$ using different integration schemes, i.e. standard RK integrators where the quaternion dynamics is respectively not stabilized (Q RK) and stabilized with Baumgarte's stabilization (Q RK+BG), and a structure-preserving integrator acting on the Lie group itself (Q RK-MK). (b), deviation of the DCM representation from $SO(3)$ using different integration schemes, i.e. standard RK integrators where the DCM dynamics is respectively not stabilized (DCM RK) and stabilized with Baumgarte's stabilization (DCM RK+BG), and a structure-preserving integrator acting on the Lie group itself (DCM RK-MK).

Since the time horizon $T = 3$s is discretized into $N = 100$ time intervals ($h = 0.03$s), the accuracy of the computed solution is on the order of $h^p = 10^{-7}$, i.e. lower by two orders of magnitude than the drift from the manifold, in those cases in which the dynamics of the $SO(3)$ parameterization is not stabilized, nor a structure-preserving integrator is employed (see curves denoted as Q RK and DCM RK in Fig. 9).

There is a significant difference between the use of a structure-preserving integration scheme and the stabilization of a standard integrator with Baumgarte's technique. In fact, while on the one hand Lie group integrators ensure consistency of the parameterization up to machine accuracy over the whole time horizon (see Q RK-MK and DCM RK-MK plots in Fig. 9), Baumgarte's stabilization only guarantees exponential convergence to the manifold in continuous time. Even though an accurate calibration of the coefficients γ in Eqs. (46) and (47) may result in a very fast convergent dynamics of the drift, there is no way of actively controlling the size of the error over time (see Fig. 9, Q RK+BG and DCM RK+BG). The error dynamics in discrete time is in fact influenced by the angular velocity ω which, in this case, is determined by the dynamics of the unicycle and acts like an exogenous signal. Moreover, it should be noted that setting the coefficients γ to large values in an attempt to make the error's dynamics as fast as possible might cause the equations to become too stiff and lose stability in discrete time.

Finally, we compared the computational efficiency of the different formulations in terms of average number of exact Newton iterations required by the solver to converge to an optimal solution. Fig. 10 shows that, in both dynamic models, the use of Baumgarte's method for the standard RK integrator leads to a noticeable increase in the number of iterations to convergence. Of course, larger values of the coefficients γ — i.e. faster dynamics of the drift — would exacerbate numerical problems and lead to a marked increase in the average number of iterations needed by the solver to converge. The best performance obtained in terms of number of iterations corresponds to the case of quaternion parameterization and the RK-MK integrator on the Lie group $SU(2)$ (see Fig. 10, Q RK-MK): here, the mean value of the number of iterations to convergence is the lowest, and also its standard deviation is considerably smaller compared to all the other formulations.

Since the computational burden per iteration is not identical for all formulations, as the amount of calculations to be performed is not the same (recall Table 3), a comparison between the average total CPU time required to converge to a solution of the OCP for the different formulations is presented in Fig. 11. The results show how the dynamic model based on quaternions (Q) is usually faster to be optimized compared to the one where the rotation is parameterized via Direction Cosine Matrix (DCM). The non-stabilized integration of both models shows the best performance, but at the cost of a non controllable drift. On the other hand, general conclusions cannot be drawn about the comparison in terms of computational efficiency between standard integrators stabilized through Baumgarte's method and structure-preserving integrators: the latter usually require a lower number of iterations to converge, albeit more computationally demanding. In terms of drift control, how-

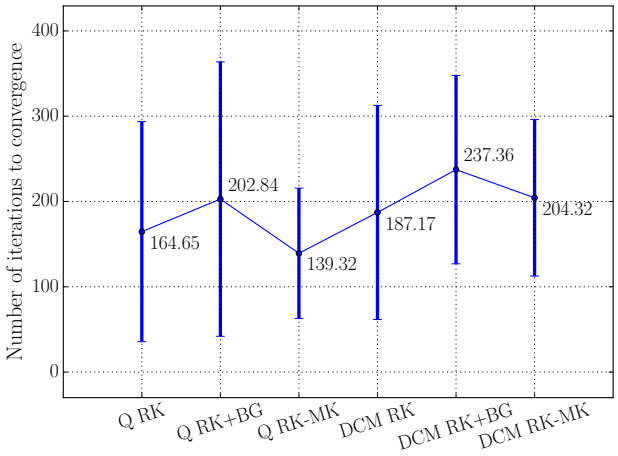


Fig. 10 Number of iterations to convergence: mean value ± standard deviation for each formulation.

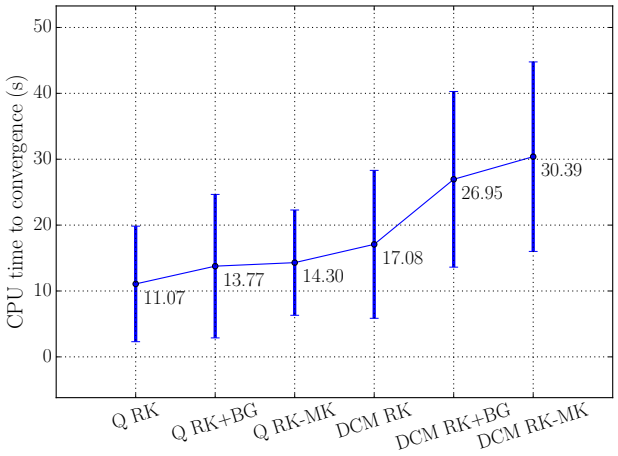


Fig. 11 CPU time to convergence: mean value ± standard deviation for each formulation.

ever, structure-preserving integrators are far superior as they guarantee the conservation of the properties of the underlying manifold up to numerical accuracy.

5.4 Analysis of the simulation accuracy

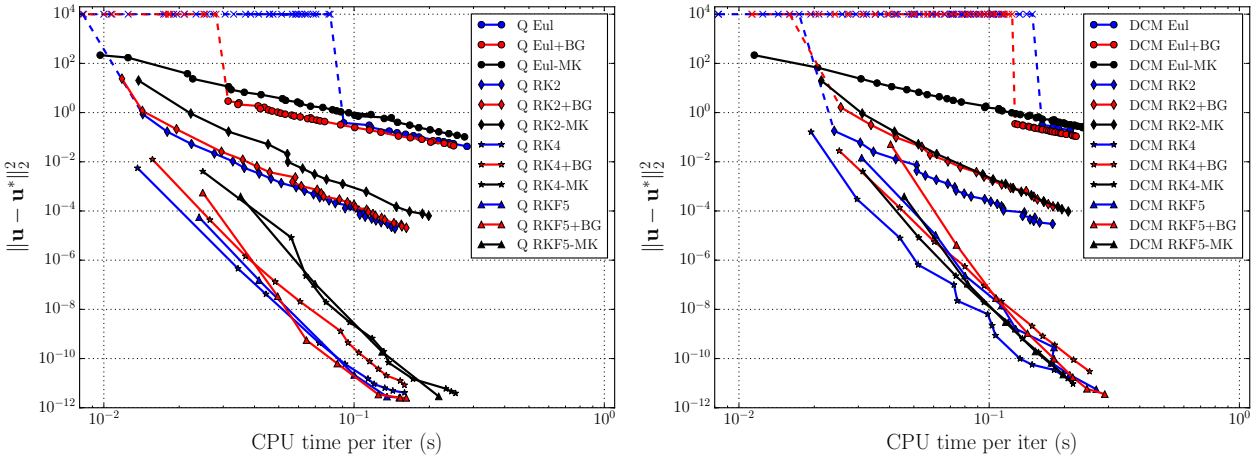
In order to analyze the effect of the *simulation accuracy* (dependent on the order of the integrator and on the number of steps) on the accuracy of the solution of the OCP, one can compare the obtained control solution trajectories with a very accurate reference solution. As a measure of the accuracy of the solution of the OCP, we define $\|\mathbf{u}-\mathbf{u}^*\|_2^2$ as *solution accuracy*, i.e. the squared L_2-norm of the difference between the computed optimal control trajectory $\mathbf{u}$ and $\mathbf{u}^*$, obtained in the reference case. In order to relate the solution accuracy to the computational effort, it is enough to perform the comparisons for one specific OCP with a fixed control and multiple-shooting grid, but with different numbers of integration steps on each interval. The task to be performed was defined by setting $x_{\text{des}} = 0.5\,\text{m}$, $y_{\text{des}} = 0.5\,\text{m}$, $\theta_{\text{des}} = \pi$. A very accurate reference solution for the corresponding OCP was then computed by discretizing the time horizon $T = 3\,\text{s}$ into $N = 25$ time intervals. The OCP was formulated by employing a DCM parameterization for the dynamics of the rotational DOF of the pendulum and a highly accurate integration scheme, i.e. a 25-step 4^{th}-order RK-MK Lie group integrator. Different solutions of the same OCP were subsequently computed by using integrators of lower accuracies. Particularly, for each formulation of the dynamics (i.e. Q and DCM), different integrators were employed:

- Standard 1^{st}- (i.e. Euler, denoted as Eul), 2^{nd}- (i.e. Heun, denoted as RK2), 4^{th}- (denoted as RK4) and 5^{th}-order (i.e. Fehlberg, denoted as RKF5) explicit RK integrators, without any stabilization for the singularity-free parameterization of $SO(3)$
- Standard 1^{st}-, 2^{nd}-, 4^{th}- and 5^{th}-order explicit RK integrators, with singularity-free parameterization of $SO(3)$ stabilized via Baumgarte's method (respectively Eul+BG, RK2+BG, RK4+BG and RKF5+BG)
- 1^{st}-, 2^{nd}-, 4^{th}- and 5^{th}-order RK-MK Lie group integrators (respectively Eul-MK, RK2-MK, RK4-MK and RKF5-MK).

The obtained results are reported in Fig. 12, where the solution accuracy corresponding to different integrators is represented as a function of the average CPU time required to perform an exact Newton iteration: for each integration scheme, the increase in simulation accuracy obtained with an increased number of integration steps per time interval obviously entails an increased computational burden. In order to avoid convergence to different local minima, all of the formulations were initialized at the reference solution. It is worth mentioning that for very low integration accuracies, standard (i.e. non structure-preserving) integrators may not enable the NLP solver to obtain a meaningful solution due to considerable drift of the non-minimal parameterization of $SO(3)$ from the corresponding manifold. Particularly, the drift of a singularity-free representation of a 3D rotation from $SO(3)$ due to a very inaccurate integration may result in a significant deformation of the pendulum, which causes the dynamic model to be poorly representative of the physical reality. In this case, the representation $\mathbf{P}$ of the actual rotation matrix $\mathbf{R} \in SO(3)$ is not orthonormal and therefore introduces some deformation. In order to exclude those non-meaningful solutions from our analysis, a polar decomposition of the rotation representation $\mathbf{P}$ was performed, which reads:

$$\mathbf{P} = \mathbf{U}\Sigma\mathbf{V}^\top = \mathbf{U}\mathbf{V}^\top\mathbf{V}\Sigma\mathbf{V}^\top = \mathbf{R}\mathbf{E} \tag{54}$$

Here, $\mathbf{U}\Sigma\mathbf{V}^\top$ is the singular value decomposition (SVD) of the representation matrix $\mathbf{P}$, whereas in its factorization



(a) Q representation

(b) DCM representation

Fig. 12 Difference between the OCP solutions computed by employing each integrator and the reference (most accurate) one, in terms of squared L_2-norm of the difference of the control trajectories as a function of the average computational time needed to perform an exact Newton iteration. For both formulations — Q in (a), DCM in (b) —, the points marked by a cross at the value 10^4 of solution accuracy and connected by a dashed line indicate that the simulation accuracy of the corresponding integrators is so poor that either the solver failed to converge or the obtained solution was clearly not in the neighborhood of the reference solution anymore.

$\mathbf{P}=\mathbf{RE}$, $\mathbf{R}=\mathbf{UV}^\top \in SO(3)$ is a unitary orthonormal matrix and the symmetric positive-definite matrix $\mathbf{E}=\mathbf{V\Sigma V}^\top$ is the deformation matrix. The elements of the main diagonal of matrix Σ are the singular values σ_i of $\mathbf{P}$. In the event that they are not unitary, the transformation $\mathbf{P}$ deforms the geometry of the rotated object, the pendulum in this case. We consider as acceptable only the solutions where such deformations are lower than a certain threshold value ε, fixed at 5%. Namely, in order for a solution of the OCP to be acceptable, the singular values of the deformations due to numerical drift from the manifold must satisfy:

$$\max_{i\in\{1,2,3\}} \sigma_i \leq 1+\varepsilon \tag{55}$$

The cases in which the solver either computed an unacceptable solution or was not able to converge to a solution at all are represented by a cross in the plots of Fig. 12.

The results show how, in both formulations, standard RK integrators most often fail to compute a physically meaningful solution for very low simulation accuracies (see curves denoted as Q Eul, DCM Eul): in such cases, the drift from the manifold is so large that even Baumgarte's stabilization is not able to properly cope with it (see Q Eul+BG, DCM Eul+BG). The only curves which seem to make sense for very low accuracies are those corresponding to the structure-preserving integrators (Q Eul-MK, DCM Eul-MK). Here, despite the low simulation accuracy, the validity of the dynamics is guaranteed by the fact that the integration scheme intrinsically preserves the structure of the singularity-free representation of $SO(3)$. For accuracies in the interval $[10^{-8},10^{-2}]$ the most efficient discretization method turns out to be the non-stabilized standard RK4 integrator with quaternion representation (Q RK4). At higher accuracies, for the Q parameterization, standard integrators stabilized with Baumgarte's method outperform the non-stabilized ones, see e.g. the Q RKF5+BG curve. In fact, when a quaternion parameterization is employed, stabilized standard integrators perform better than the structure-preserving ones, see Fig. 12 (a). Interestingly, the results reported in Fig. 12 (b) show that the same does not hold when a DCM representation is employed. In this case, in fact, the use of standard integrators with Baumgarte's stabilization or of RK-MK integrators lead to a comparable computational performance. Structure-preserving integrators seem to be slightly more competitive for higher accuracies (see DCM RK4-MK, DCM RKF5-MK plots). Moreover, for the DCM representation, standard RK methods without stabilization are the most convenient for accuracies up to 10^{-10}: due to the lower complexity of the resulting dynamic model, they are the least computationally demanding.

6 Conclusions

In this paper, the application of structure-preserving integration schemes to direct optimal control has been investigated. Two Munthe-Kaas Lie group integrators have been implemented and applied to the time integration of redundant parameterizations of a 3D rotation in the context of the numerical solution of an optimal control problem. Their performance has been compared to that of standard integrators in terms of drift from the manifold, solution accuracy, qual-

ity of the optimal trajectory, and computational efficiency to convergence.

Overall, the quaternion-based representation turned out to be the most efficient both in terms of iterations and CPU time to convergence, but at the cost of lower success rates and increased probability of being trapped by higher local minima. When a quaternion parameterization is employed, stabilized standard integrators perform better than the structure-preserving ones. On the other hand, when a DCM representation is adopted, standard integrators with Baumgarte's stabilization exhibit a computational performance comparable to that of RK-MK schemes. Structure-preserving integrators are the only choice for lower simulation accuracies, whereas higher-order, non-stabilized standard integrators seem to be most competitive when higher levels of solution accuracy are pursued.

Acknowledgements This work was partially supported by Grant No. 645599 "SoMa" (Soft-bodied intelligence for Manipulation) within the H2020-ICT-2014-1 program. Support by the EU via ERC-HIGHWIND (259 166), ITN-TEMPO (607957) and ITN-AWESCO (642 682), by DFG via Research Unit FOR 2401, and by the German BMWi via the project eco4wind is also gratefully acknowledged.

Marco Gabiccini wishes to thank the colleagues Luca Greco and Paolo Mason from the Laboratoire des Signaux & Systèmes, CentraleSupélec for having initially disclosed the benchmark mechanical system investigated in this paper.

References

1. Hairer, E., Lubich, C., Wanner, G.: Geometric numerical integration: structure-preserving algorithms for ordinary differential equations, vol. 31. Springer Science & Business Media (2006)
2. Brüls, O., Cardona, A.: On the use of Lie group time integrators in multibody dynamics. Journal of Computational and Nonlinear Dynamics **5**(3), 031,002 (2010)
3. Betsch, P.: Structure-preserving Integrators in Nonlinear Structural Dynamics and Flexible Multibody Dynamics. Courses and Lectures. Springer (2016)
4. Stuelpnagel, J.: On the parametrization of the three-dimensional rotation group. SIAM review **6**(4), 422–430 (1964)
5. Shuster, M.D.: A survey of attitude representations. Navigation **8**(9), 439–517 (1993)
6. Marsden, J.E., Ratiu, T.: Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems, vol. 17. Springer Science & Business Media (2013)
7. Gros, S., Zanon, M., Diehl, M.: Baumgarte stabilisation over the SO (3) rotation group for control. In: 2015 54th IEEE Conference on Decision and Control (CDC), pp. 620–625. IEEE (2015)
8. Baumgarte, J.: Stabilization of constraints and integrals of motion in dynamical systems. Computer Methods in Applied Mechanics and Engineering **1**(1), 1–16 (1972)
9. Iserles, A., Munthe-Kaas, H.Z., Nørsett, S.P., Zanna, A.: Lie-group methods. Acta Numerica 2000 **9**, 215–365 (2000)
10. Celledoni, E., Owren, B.: Lie group methods for rigid body dynamics and time integration on manifolds. Computer Methods in Applied Mechanics and Engineering **192**(3), 421–438 (2003)
11. Engø, K., Marthinsen, A.: Modeling and solution of some mechanical problems on Lie groups. Multibody System Dynamics **2**(1), 71–88 (1998)
12. Park, J., Chung, W.K.: Geometric integration on Euclidean group with application to articulated multibody systems. IEEE Transactions on Robotics **21**(5), 850–863 (2005)
13. Bottasso, C.L., Borri, M.: Integrating finite rotations. Computer Methods in Applied Mechanics and Engineering **164**(3), 307–331 (1998)
14. Crouch, P.E., Grossman, R.: Numerical integration of ordinary differential equations on manifolds. Journal of Nonlinear Science **3**(1), 1–33 (1993)
15. Munthe-Kaas, H.: Runge-Kutta methods on Lie groups. BIT Numerical Mathematics **38**(1), 92–111 (1998)
16. Terze, Z., Müller, A., Zlatar, D.: Lie-group integration method for constrained multibody systems in state space. Multibody System Dynamics **34**(3), 275–305 (2015)
17. Terze, Z., Müller, A., Zlatar, D.: Singularity-free time integration of rotational quaternions using non-redundant ordinary differential equations. Multibody System Dynamics **38**(3), 201–225 (2015)
18. Betts, J.T.: Practical methods for optimal control and estimation using nonlinear programming, 2nd edn. SIAM (2010)
19. Bottasso, C.L., Croce, A.: Optimal control of multibody systems using an energy preserving direct transcription method. Multibody system dynamics **12**(1), 17–45 (2004)
20. Chung, C.Y., Lee, J.W., Lee, S.M., Lee, B.H.: Balancing of an inverted pendulum with a redundant direct-drive robot. In: Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on, vol. 4, pp. 3952–3957. IEEE (2000)
21. Yang, R., Kuen, Y.Y., Li, Z.: Stabilization of a 2-DOF spherical pendulum on xy table. In: Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on, pp. 724–729. IEEE (2000)
22. Albouy, X., Praly, L.: On the use of dynamic invariants and forwarding for swinging up a spherical inverted pendulum. In: Decision and Control, 2000. Proceedings of the 39th IEEE Conference on, vol. 2, pp. 1667–1672.

IEEE (2000)
23. Liu, G., Nešić, D., Mareels, I.: Modelling and stabilisation of a spherical inverted pendulum. IFAC Proceedings Volumes **38**(1), 1130–1135 (2005)
24. Do, K.D., Seet, G.: Motion control of a two-wheeled mobile vehicle with an inverted pendulum. Journal of Intelligent & Robotic Systems **60**(3), 577–605 (2010)
25. Jung, S., Kim, S.S.: Control experiment of a wheel-driven mobile inverted pendulum using neural network. IEEE Transactions on Control Systems Technology **16**(2), 297–303 (2008)
26. Yue, M., Wei, X., Li, Z.: Adaptive sliding-mode control for two-wheeled inverted pendulum vehicle based on zero-dynamics theory. Nonlinear Dynamics **76**(1), 459–471 (2014)
27. Yoon, M.G.: Dynamics and stabilization of a spherical inverted pendulum on a wheeled cart. International Journal of Control, Automation and Systems **8**(6), 1271–1279 (2010)
28. Shabana, A.A.: Computational dynamics. John Wiley & Sons (2009)
29. Shabana, A.A.: Dynamics of multibody systems, 4th edn. Cambridge University Press (2013)
30. Holm, D.D.: Geometric Mechanics. Part II: Rotating, Translating and Rolling. Imperial College Press, London (2011)
31. Nocedal, J., Wright, S.: Numerical optimization. Springer Verlag, New York (2006)
32. Sternberg, J., Gros, S., Houska, B., Diehl, M.: Approximate robust optimal control of periodic systems with invariants and high-index differential algebraic systems. IFAC Proceedings Volumes **45**(13), 690–695 (2012)
33. Betts, J.T.: Survey of numerical methods for trajectory optimization. Journal of Guidance, Control, and Dynamics **21**(2), 193–207 (1998)
34. Bock, H.G., Plitt, K.J.: A multiple shooting algorithm for direct solution of optimal control problems. Proceedings of the IFAC World Congress (1984)
35. Bryson, A.E., Ho, Y.C.: Applied Optimal Control. John Wiley & Sons, New York (1975)
36. Hager, W.W.: Runge-Kutta methods in optimal control and the transformed adjoint system. Numerische Mathematik **87**(2), 247–282 (2000)
37. Gong, Q., Ross, I.M., Kang, W., Fahroo, F.: On the pseudospectral covector mapping theorem for nonlinear optimal control. In: Decision and Control, 2006 45th IEEE Conference on, pp. 2679–2686. IEEE (2006)
38. Hairer, E., Nørsett, S., Wanner, G.: Solving ordinary differential equations I: nonstiff problems, vol. 8. Springer Series in Computational Mathematics (1993)
39. Ascher, U.M.: Stabilization of invariants of discretized differential systems. Numerical Algorithms **14**(1-3), 1–24 (1997)
40. Erhard, M., Horn, G., Diehl, M.: A quaternion-based model for optimal control of the SkySails airborne wind energy system. Zeitschrift für Angewandte Mathematik und Mechanik **97**(1), 7–24 (2017). DOI 10.1002/zamm.201500180
41. Gros, S., Zanon, M., Vukov, M., Diehl, M.: Nonlinear MPC and MHE for mechanical multi-body systems with application to fast tethered airplanes. IFAC Proceedings Volumes **45**(17), 86–93 (2012)
42. Andrle, M.S., Crassidis, J.L.: Geometric integration of quaternions. Journal of Guidance, Control, and Dynamics **36**(6), 1762–1767 (2013)
43. Murray, R.M., Li, Z., Sastry, S.S.: A mathematical introduction to robotic manipulation. CRC Press (1994)
44. Andersson, J.: A General-Purpose Software Framework for Dynamic Optimization. PhD thesis, Arenberg Doctoral School, Katholieke Universiteit Leuven (2013)
45. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming **106**(1), 25–57 (2006)
46. Griewank, A., Walther, A.: Evaluating derivatives: principles and techniques of algorithmic differentiation. SIAM (2008)
47. Lago Garcia, J.: Periodic Optimal Control and Model Predictive Control of a Tethered Kite for Airborne Wind Energy. Master's thesis, University of Freiburg (2016)
48. Manara, S., Gabiccini, M., Artoni, A., Diehl, M.: On the integration of singularity-free representations of SO(3) for direct optimal control — submission accompanying video. https://youtu.be/XDRX3w40oaQ (2017)