Robotics and Autonomous Systems 98 (2017) 347–359

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

# Point stabilization of nonholonomic spherical mobile robot using nonlinear model predictive control

Mahmood Reza Azizi, Jafar Keighobadi *

*University of Tabriz, Faculty of Mechanical Engineering, 29 Bahman, Tabriz, P.C.5166614766, Iran*

## HIGHLIGHTS

- A spherical mobile robot driven by two perpendicular rotors is investigated.
- The equations of motion are derived using angular momentum conservation principle.
- The controllability of the model is analyzed and its uncontrollable configurations are specified.
- A NMPC is designed to control the position and orientation of the spherical shell simultaneously.
- The stability of designed controller is proven and its performance is evaluated.

## ARTICLE INFO

*Article history:*
Received 12 April 2017
Received in revised form 20 July 2017
Accepted 23 September 2017
Available online 4 October 2017

*Keywords:*
Spherical mobile robot
Nonholonomic system
Model predictive control
Point stabilization

## ABSTRACT

Control of nonholonomic spherical mobile robot is a generalization of the classical ball-plate problem which is still challenging in robotic researches. In this paper, point stabilization of a nonholonomic spherical mobile robot actuated by two internal rotors is investigated. Since every kinematic trajectory is not always dynamically realizable for the spherical robot driven by two actuators, the mathematical model of the robot is derived based on the angular momentum conservation principle. The controllability of the robot is evaluated based on the obtained model and the uncontrollable configurations as well as their geometrical meaning are specified. To simultaneous control of position and orientation of the robot, a nonlinear model predictive control (NMPC) is developed for the first time and the stability analysis is performed through using Lyapunov stability theorem. The performance of the designed control system is assessed through computer simulations in different test conditions. The simulation results show the significant performance of the proposed NMPC in stabilization of the spherical shell from every initial configuration to every desired position and orientation even in the uncontrollable region. Considering additive bounded noises, the robust stabilization of the nonholonomic spherical robot by the NMPC is also assessed in simulations.



## 1. Introduction

Nonholonomy is a characteristic of mechanical systems that makes it possible to control multiple state variables with fewer number of control inputs. As an interesting property in design of mobile robots, the nonholonomy could help to use simpler mechanisms in producing locomotion and mobility [1]. The spherical mobile robot comprising a spherical shell and an inner driver mechanism is a well-known example of nonholonomic systems attracted many attentions in the past decades. Compared to the classical wheeled and legged mobile robots, the spherical mobile robot provides special advantages by its shape and structure. Omnidirectional locomotion with merely two or three actuators, stable locomotion even after collision or falling from height, smooth outer shape and protective spherical shield for the inner parts are some of its inherent advantages that discriminate the spherical robot from the other types of mobile robots [2].

While many studies have focused on mechanical design and dynamic modeling of the spherical robot [3–7], one of the major problems with this type of mobile robots is its stabilization and control system design which is still an open problem. The spherical robot is restricted by nonholonomic constraints due to its perfect rolling motion. On the other hand, according to Brockett's theorem [8], nonholonomic systems do not meet the necessary conditions for feedback stabilization and any smooth time invariant state feedback does not exist to stabilize such systems on the equilibrium point. As a result, the classical nonlinear control approaches like

* Corresponding author.
*E-mail addresses:* mhr_azizi@yahoo.com (M.R. Azizi), keighobadi@tabrizu.ac.ir (J. Keighobadi).

https://doi.org/10.1016/j.robot.2017.09.015
0921-8890/

the feedback linearization are not adequate to control all states of the systems. Furthermore, a systematic control strategy has not been still reported for stabilization of general nonholonomic systems. The current methods in the literature are limited to some special classes of nonholonomic systems such as chained form, power form and differentially flat systems [9–12]. Therefore, for motion control of the spherical robot several investigations have been reported in the recent literature that all of them could be classified into either straight-line motion or curvilinear motion control.

The control and stabilization of the spherical robot in straight line motions have been studied by several researchers. Considering the uncertainties and disturbances in the robot model, the stabilization of spherical robot in linear motion has been investigated using input-state linearization, adaptive hierarchical sliding mode approach, adaptive neuro-fuzzy sliding mode controller, neural network controller, linear control methods and gray-PID type fuzzy controller [13–20]. However, the robot constraints in straight line motion imposed by the perfect rolling condition could be integrated and converted to holonomic constraints. Hence, the straight-line motion control does not involve too much complexities of general nonholonomic situation.

To stabilize the spherical robot in every desired position and orientation, the robot's motion on curvilinear trajectories should be investigated. Bhattacharya et al. have shown that the spherical robot actuated by two perpendicular rotors is a controllable nonholonomic system [21]. Since the robot system is neither flat nor nilpotent and could not be converted to chained form, the classical nonlinear control approaches could not be utilized for its stabilization [21]. Similar results have also been represented by Joshi et al. for kinematic model of the robot obtained using unit quaternion [22].

Several studies have been conducted recently for curvilinear motion control of the spherical robot driven by two actuators. However, to decrease the technical complexities, some simplifying assumptions have been considered. In most of the preceding researches, the trajectory planning and controller design problems have been performed based on the kinematical model of the robot and by neglecting the angular velocity component of the spherical shell along the normal direction to actuators' plane. Based on the kinematic model, the trajectory planning algorithm has been proposed using geometrical method in [23,24] and using dynamic programming approach in [25,26]. PD and adaptive sliding mode controllers have been designed to follow a trajectory on the contact plane with constant velocity [27]. Cai et al. have used the kinematic model of the robot and the shunting model of neurodynamics to control the position of the spherical shell on the horizontal plane [28]. The dynamic model of the robot has been used for position control of the spherical shell using back-stepping method, feedback linearization and back-stepping together with fuzzy guidance scheme in [29–31]. However, the rotation of the spherical shell along the normal direction to the plane of actuators has also been neglected and the controller could only steer some of the system's state variables.

In some other researches, the trajectory planning and control of the spherical robot driven by three actuators have been considered. Neglecting the dynamical effects of the inner mechanism, point stabilization of a spherical robot actuated by 3-DOF gyro has been performed using Lyapunov direct method and back-stepping approach [32]. Following developing the dynamic model of a new spherical robot equipped with a gyro and three actuators, Urakubo et al. have designed a control system merely for the translational velocity of the spherical shell based on Lyapunov stability theorem [33]. Smooth stabilizing controller for position and reduced attitude has been proposed for a spherical robot actuated by three internal rotors [34]. Dynamic modeling and motion control of the spherical robot driven by three internal rotors have been presented by Karimpour et al. using geometric mechanics perspective and Lyapunov control approach [35]. Geometric control method has been used for separately control of position and attitude of a spherical robot actuated by three internal rotors [36]. However, the spherical robot with three actuators is a nonholonomic differentially flat system and its trajectory planning and control system design are not too much complicated [37,38].

According to the above literature review, the preceding researches on trajectory planning and control of the spherical robot driven by two actuators have been performed based on the kinematic model of the robot and by neglecting the angular velocity of the spherical shell in the direction normal to actuators' plane. However, Morinaga et al. have shown that for a spherical robot driven by two actuators, any feasible kinematic trajectory is not always dynamically realizable [37]. On the other hand, the rotation of the spherical shell in the direction normal to actuators plane could not be neglected as shown in [38]. Furthermore, in the previous studies the uncontrollable configurations of the robot have not been considered in control system design. However, to the best of our knowledge, an efficient control strategy in the uncontrollable region of simultaneous position and orientation control of the robot considering dynamic realizability conditions and the angular velocity of the spherical shell in all directions has not been reported in the previously documented researches.

Therefore, the main contribution of this paper is simultaneous position and orientation control of the spherical robot actuated by two internal rotors considering the dynamic realizability condition of motion. The mathematical model of the robot is extracted using angular momentum conservation principle and quaternion algebra without neglecting the angular velocity of the spherical shell in any direction. The controllability of the robot is analyzed using the obtained model and its uncontrollable configurations are specified and their geometrical interpretation is precisely described. To move the spherical robot from every initial configuration to every desired position and orientation, the robot's mathematical model is discretized using a linear approximation for the control inputs in each sampling period (first order hold sampling (FOH)). A nonlinear model predictive control (NMPC) is designed based on the discretized model for the first time and its ingredients are determined through using Lyapunov stability theorem. Using this type of discretization, a continuous time-varying feedback control law is obtained which satisfies Brockett's necessary conditions for stabilization of the nonholonomic systems. Software simulations of the proposed control strategy show perfect stabilization of the robot even in the uncontrollable region and under Gaussian white noises affecting the robot.

## 2. Mathematical modeling

### 2.1. Kinematic model

Fig. 1a demonstrates the spherical robot driven by two perpendicular rotors. The rotors are installed inside the spherical shell and two counterweights are added to balance the rotors' weights in such a way that the robot's center of gravity coincides with the geometric center of the spherical shell. Consequently, the gravitational effects are eliminated and the locomotion could be generated based on the angular momentum conservation principle. Furthermore, the rotation axes of the rotors are assumed to be coincide with their corresponding axes of symmetry. When the rotors spin, the spherical shell rolls on the surface and its position and orientation are continuously changed. This motion is assumed to be rolling without slipping. Fig. 1b illustrates three coordinate frames used to describe the position and orientation of the spherical shell. Frame {1} attached to the contact plane is an inertial

reference frame. Frame {2} is attached to the geometric center of the spherical shell and its axes remain parallel to the axes of the inertial reference frame. Frame {3} is a body coordinate frame attached to the geometric center of the spherical shell and its $x$- and $y$-axes are aligned with the axes of rotation of the internal rotors. Let $(x, y)$ be the position vector components of the center of the spherical shell with respect to frame {1} and $(q_1,\ q_2,\ q_3,\ q_4)$ be the components of a unit quaternion, the position and orientation of the spherical shell with respect to frame {1} could be described by these six generalized coordinates. The coordinate transformation operator of frame {3} with respect to frame {2} is defined using quaternion algebra in the following form [39,40].

$$ {}^2_3L_{\mathbf{q}}(\mathbf{P}) = \mathbf{q} \otimes \begin{bmatrix} 0 \\ \mathbf{P} \end{bmatrix} \otimes \bar{\mathbf{q}}; \quad {}^3_2L_{\mathbf{q}} = {}^2_3L_{\bar{\mathbf{q}}} \tag{1} $$

$$ \begin{aligned} &\mathbf{q} = [q_1 \quad q_2 \quad q_3 \quad q_4]^T = \left[\cos\left(\frac{\psi}{2}\right) \quad \mathbf{v}\sin\left(\frac{\psi}{2}\right)\right]^T \\ &\|\mathbf{q}\|^2 = q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \end{aligned} \tag{2} $$

where, $\otimes$ stands for quaternions multiplication symbol, $\mathbf{P} \in \mathbb{R}^3$ is an arbitrary vector in {3}, $\psi$ is the angle of rotation and $\mathbf{v}$ is a unit vector along the axis of rotation. The angular velocity of the spherical shell is obtained as follows.

$$ \begin{bmatrix} 0 \\ {}^2\boldsymbol{\omega}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ {}^2\boldsymbol{\omega}_S \end{bmatrix} = 2\dot{\mathbf{q}} \otimes \bar{\mathbf{q}} \tag{3} $$

$$ {}^2\boldsymbol{\omega}_3 = {}^2\boldsymbol{\omega}_S = \begin{bmatrix} -q_2 & q_1 & -q_4 & q_3 \\ -q_3 & q_4 & q_1 & -q_2 \\ -q_4 & -q_3 & q_2 & q_1 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} \tag{4} $$

$$ \begin{bmatrix} 0 \\ {}^3\boldsymbol{\omega}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ {}^3\boldsymbol{\omega}_S \end{bmatrix} = {}^3_2L_{\mathbf{q}}\left({}^2\boldsymbol{\omega}_3\right) \tag{5} $$

The angular velocity of the rotors and counterweights are also determined as follows.

$$ \begin{aligned} &{}^3\boldsymbol{\omega}_{r1} = {}^3\boldsymbol{\omega}_S + \begin{bmatrix} u_1 \\ 0 \\ 0 \end{bmatrix}; \quad {}^3\boldsymbol{\omega}_{r2} = {}^3\boldsymbol{\omega}_S + \begin{bmatrix} 0 \\ u_2 \\ 0 \end{bmatrix}; \\ &{}^3\boldsymbol{\omega}_{cw1} = {}^3\boldsymbol{\omega}_{cw2} = {}^3\boldsymbol{\omega}_S \end{aligned} \tag{6} $$

where, the considered control inputs, $u_1$ and $u_2$ are the angular velocity of the rotors with respect to the spherical shell. Since the robot's motion is assumed to be rolling without slipping, the velocity of the contact point between the spherical shell and ground is zero.

$$ {}^2\mathbf{V}_O = {}^2\mathbf{V}_I + {}^2\boldsymbol{\omega}_3 \times R_S\,\hat{\mathbf{k}}_2 = {}^2\boldsymbol{\omega}_3 \times R_S\,\hat{\mathbf{k}}_2 $$

$$ \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = 2R_S \begin{bmatrix} -q_3\dot{q}_1 + q_4\dot{q}_2 + q_1\dot{q}_3 - q_2\dot{q}_4 \\ q_2\dot{q}_1 - q_1\dot{q}_2 + q_4\dot{q}_3 - q_3\dot{q}_4 \end{bmatrix} \tag{7} $$

Eq. (7) includes two non-integrable relations between the generalized coordinates and velocities which describe two nonholonomic constraints originated from non-slipping motion of the robot.

### 2.2. Equations of motion

The kinematic model obtained in the previous section does not completely describe the behavior of the system under consideration. Indeed, due to rolling without slipping motion, the resultant of the external moments about the contact point between the spherical shell and the ground is zero. Therefore, the angular momentum of the whole system about this point will be conserved during the robot's motion. The angular momentum of the whole system about the robot's center of gravity could be calculated in the following form.

$$ \begin{aligned} {}^3\mathbf{H}_O &= {}^3\mathbf{H}_O^S + {}^3\mathbf{H}_O^{r1} + {}^3\mathbf{H}_O^{r2} + {}^3\mathbf{H}_O^{cw1} + {}^3\mathbf{H}_O^{cw2} = \\ &\left(\mathbf{I}_S + \mathbf{I}_{cw1} + \mathbf{I}_{cw2} + m_{cw}\begin{bmatrix} 0 & 0 & 0 \\ 0 & d^2 & 0 \\ 0 & 0 & d^2 \end{bmatrix} + m_{cw}\begin{bmatrix} d^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & d^2 \end{bmatrix}\right){}^3\boldsymbol{\omega}_S \\ &+ \left(\mathbf{I}_{r1} + m_{r1}\begin{bmatrix} 0 & 0 & 0 \\ 0 & d^2 & 0 \\ 0 & 0 & d^2 \end{bmatrix}\right){}^3\boldsymbol{\omega}_{r1} \\ &+ \left(\mathbf{I}_{r2} + m_{r2}\begin{bmatrix} d^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & d^2 \end{bmatrix}\right){}^3\boldsymbol{\omega}_{r2} \end{aligned} \tag{8} $$

where, the subscripts and superscripts $S$, $cw$, $r$ denote spherical shell, counterweights and rotors respectively; $\mathbf{I}$ denotes the moment of inertia matrix of each part about its center of gravity; $m$ is the mass of each part and the distance, $d$ is shown in Fig. 1a. The moment of inertial of every part of the robot is calculated about the robot's center of gravity in {3} using the parallel axes theorem. The vector of angular momentum about the contact point, $\mathbf{P}_I$ is obtained as follows.

$$ \begin{bmatrix} 0 \\ {}^3\mathbf{H}_{\mathbf{P}_I} \end{bmatrix} = \begin{bmatrix} 0 \\ {}^3\mathbf{H}_O \end{bmatrix} + {}^3_2L_{\mathbf{q}}\left(R_S\hat{\mathbf{k}}_2 \times {}^2\mathbf{L}\right) \tag{9} $$

where, the total linear momentum vector of the robot in frame {2}, ${}^2\mathbf{L}$ is determined as:

$$ \begin{aligned} &{}^2\mathbf{L} = M_t R_S \begin{bmatrix} -q_3\dot{q}_1 + q_4\dot{q}_2 + q_1\dot{q}_3 - q_2\dot{q}_4 \\ q_2\dot{q}_1 - q_1\dot{q}_2 + q_4\dot{q}_3 - q_3\dot{q}_4 \\ 0 \end{bmatrix} \\ &M_t = M_S + 2m_r + 2m_{cw} \end{aligned} \tag{10} $$

Assuming that the robot is initially at rest and its motion is started from zero velocity, the angular momentum of the whole system around the contact point is initially zero and remains constant during the robot's motion. Therefore, the differential equations of motion are obtained as follows.

$$ {}^3\mathbf{H}_{\mathbf{P}_I} = 0 \tag{11} $$

Eq. (11) is a set of three first order nonlinear differential equations in the following form.

$$ \begin{aligned} &2I_{t1}\left(-\dot{q}_1q_2 + \dot{q}_2q_1 + \dot{q}_3q_4 - \dot{q}_4q_3\right) + I_1\,u_1 \\ &\quad + 2M_tR_S\dot{x}\left(q_1q_4 + q_2q_3\right) + M_tR_S\dot{y}\left(q_3^2 + q_4^2 - q_1^2 - q_2^2\right) = 0 \\ &2I_{t1}\left(-\dot{q}_1q_3 - \dot{q}_2q_4 + \dot{q}_3q_1 + \dot{q}_4q_2\right) + I_1\,u_2 \\ &\quad + 2M_tR_S\dot{y}\left(q_1q_4 - q_2q_3\right) + M_tR_S\dot{x}\left(q_1^2 + q_3^2 - q_2^2 - q_4^2\right) = 0 \\ &M_tR_S\left[\dot{x}\left(q_1q_2 - q_3q_4\right) + \dot{y}\left(q_1q_3 + q_2q_4\right)\right] \\ &\quad + I_{t\,2}\left(-\dot{q}_1q_4 + \dot{q}_2q_3 - \dot{q}_3q_2 + \dot{q}_4q_1\right) = 0 \\ &I_{t1} = I_1 + I_{c1} + \left(I_2 + m_rd^2\right) + \left(I_{c2} + m_{cw}d^2\right) + I_S \\ &I_{t2} = 2\left(I_2 + m_rd^2\right) + 2\left(I_{c2} + m_{cw}d^2\right) + I_S \end{aligned} \tag{12} $$

where, $I_1$ and $I_{c1}$ stand for the moments of inertia of the rotors and the counterweights around their axes of symmetry; similarly $I_2$ and $I_{c2}$ are the same moments of inertia around two remainder perpendicular axes. Eq. (12) expresses the conservation of the total angular momentum vector of the robot described in frame {3} about the contact point of the spherical shell with the ground. It is obvious that, the third equation of (12) is a non-integrable relation between the generalized coordinates and velocities. This equation describes another nonholonomic constraint originated

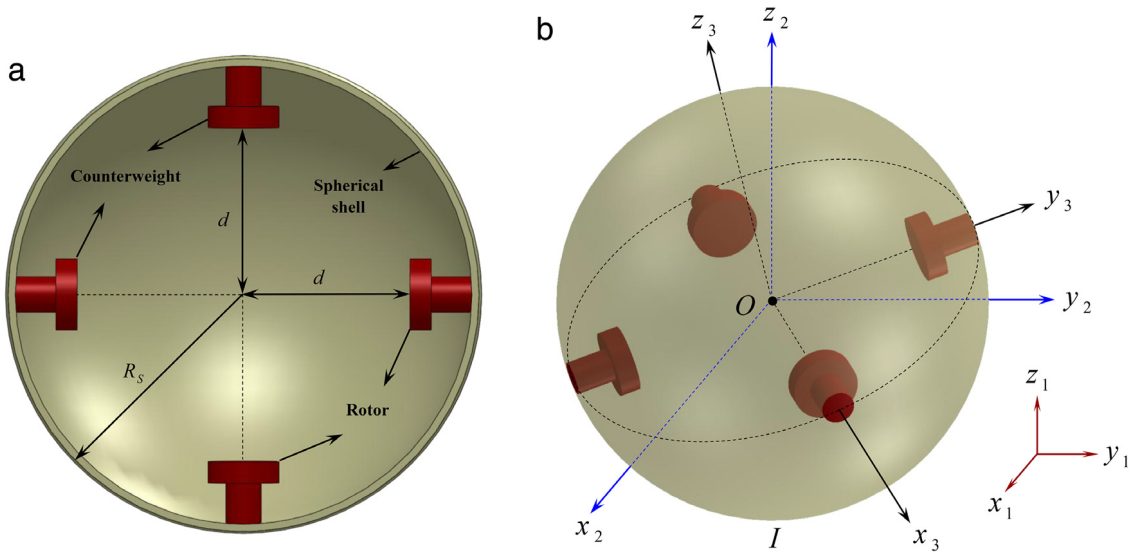


**Fig. 1.** Spherical robot, a: the robot's construction, b: the attached coordinate frames.

from the conservation of the angular momentum and the lack of actuator inside the spherical shell in the $z$ direction of frame {3}. In addition, the third equation of (12), which could be considered as the dynamic realizability constraint, is a relation between the angular velocity of the spherical shell in $z_3$ direction (${}^3\omega_{3z} = -\dot{q}_1 q_4 + \dot{q}_2 q_3 - \dot{q}_3 q_2 + \dot{q}_4 q_1$) and linear velocity of the robot and other generalized coordinates. This shows that the angular velocity of the spherical shell along $z_3$ direction is not always zero.

### 2.3. State space representation

To obtain the state space representation of the system under consideration, the time derivative of (2) gives:

$$q_1\dot{q}_1 + q_2\dot{q}_2 + q_3\dot{q}_3 + q_4\dot{q}_4 = 0 \tag{13}$$

Now, the differential equations of motion (12) obtained in the earlier section are gathered with (13) and the nonholonomic constraints (7) in the following matrix form.

$$\mathbf{Q}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\\ \dot{q}_4\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix} u_1 + \begin{bmatrix}0\\0\\0\\1\\0\\0\end{bmatrix} u_2 \tag{14}$$

where, the components of matrix $\mathbf{Q} \in \mathbb{R}^{6\times 6}$ are nonlinear functions of the generalized coordinates. Through multiplying the both sides of (14) by $\mathbf{Q}^{-1}$, the state space representation of the robot is obtained in the following affine form.

$$\dot{\mathbf{x}} = \mathbf{g}_1(\mathbf{x})\, u_1 + \mathbf{g}_2(\mathbf{x})\, u_2 = \mathbf{G}(\mathbf{x})\, \mathbf{u} \tag{15}$$

where, $\mathbf{x} : \mathbb{R}_{0+} \to \mathbb{R}^2 \times \mathbb{S}^4_1$ is the vector of state variables; $\mathbf{u} : \mathbb{R}_{0+} \to \mathbf{U} \subset \mathbb{R}^2$ is the vector of control inputs; $\mathbf{g}_1(\cdot), \mathbf{g}_2(\cdot) : \mathbb{R}^2 \times \mathbb{S}^4_1 \to \mathbb{R}^6$ are nonlinear vector functions of states; $\mathbf{G}(\cdot) : \mathbb{R}^2 \times \mathbb{S}^4_1 \to \mathbb{R}^{6\times 2}$ is a matrix composed of $\mathbf{g}_1(\cdot)$ and $\mathbf{g}_2(\cdot)$ and $\mathbb{S}^n_\rho$ denotes a spherical surface of radius $\rho$ around the origin of $\mathbb{R}^n$.

## 3. Controller design

### 3.1. Controllability analysis

To design a control system for stabilization of the spherical robot, it is essential to determine the controllable domain of the nonlinear system (15). Therefore, the controllability of the system (15) is evaluated through the following theorem based on the controllability rank condition for the nonlinear systems [41].

**Theorem 1.** *The driftless nonlinear system (15) is controllable everywhere except where the contact point between the spherical shell and the ground lies on the great circle encompassing two internal rotors (Circle $\Gamma$, Fig. 2).*

**Proof.** Form the following distributions:

$$\Delta_1 = span\{\mathbf{g}_1, \mathbf{g}_2\}$$
$$\Delta_2 = \Delta_1 + span\{[\mathbf{g}_1, \mathbf{g}_2]\}$$
$$\Delta_3 = \Delta_C = \Delta_2 + span\{[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]], [\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]\}$$

where $[\mathbf{f}, \mathbf{g}] = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}\mathbf{f} - \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{g}$ denotes Lie bracket of two vector fields $\mathbf{f}$ and $\mathbf{g}$. Recall that the system subjects to a holonomic constraint (2); the mathematical model (15) is controllable where the rank of accessibility distribution $\boldsymbol{\Delta}_{C(6\times 5)}$ is five. The calculation is performed in Maple 18 and omitted here due to space limitation. However, one can easily check that the distribution $\boldsymbol{\Delta}_C$ is full rank everywhere except where the following relation holds between the quaternion components.

$$q_2^2 + q_3^2 = \frac{1}{2} \quad \Rightarrow \quad q_2 = \pm\sqrt{\frac{1}{2} - q_3^2} \tag{16}$$

Substituting the (16) in (2) yields:

$$q_1^2 + q_4^2 = \frac{1}{2} \quad \Rightarrow \quad q_1 = \pm\sqrt{\frac{1}{2} - q_4^2} \tag{17}$$

To explain the geometrical meaning of the above relations, the position vector of the contact point between the spherical shell and the ground is calculated in frames {2} and {3}.

$${}^2\mathbf{P}_I = \begin{bmatrix}0 & 0 & -R_S\end{bmatrix}^T$$

The position of the contact point in {3} is determined using coordinate transformation operator (1).

$$\begin{bmatrix}0\\ {}^3\mathbf{P}_I\end{bmatrix} = {}^3_2L_q\left({}^2\mathbf{P}_I\right) = \begin{bmatrix}0\\ {}^3x\\ {}^3y\\ {}^3z\end{bmatrix} = \begin{bmatrix}0\\ 2R_S(q_1q_3 - q_2q_4)\\ -2R_S(q_1q_2 + q_3q_4)\\ R_S\left(q_2^2 + q_3^2 - q_1^2 - q_4^2\right)\end{bmatrix} \tag{18}$$

Substituting (16) and (17) in (18) leads to the following equation.

$$ {}^3\mathbf{P}_I = \begin{bmatrix} {}^3x \\ {}^3y \\ {}^3z \end{bmatrix} = \begin{bmatrix} 2R_S\left(\pm q_3\sqrt{\frac{1}{2}-q_4^2} \mp q_4\sqrt{\frac{1}{2}-q_3^2}\right) \\ -2R_S\left(\sqrt{\frac{1}{2}-q_3^2}\sqrt{\frac{1}{2}-q_4^2} - q_3q_4\right) \\ 0 \end{bmatrix} \tag{19} $$

It is clear that, ${}^3x^2 + {}^3y^2 = R_S^2$ and therefore, the locus of the uncontrollable points on the sphere is the great circle in the $x$-$y$ plane of frame {3} as stated in Theorem 1. ■

**Corollary 1.** *Every trajectory of system (15) is such that the locus of the contact points on the sphere is perpendicular to the great circle (19) (Circle $\Gamma$, Fig. 2) at their intersections.*

**Proof.** Let **P** be an arbitrary point on the great circle $\Gamma$. The location of this point in coordinate frame {3} could be defined by angle $\phi$ from $x_3$ axis. When the spherical shell rotates 90° around the unit vector **n** from its initial orientation depicted in Fig. 2, the point **P** contacts with the ground and the robot is in one of its uncontrollable configurations. Other uncontrollable configurations could be achieved by spinning the spherical shell by an angle $\alpha$ $(0 \le \alpha \le 2\pi)$ around the vertical axis. The quaternions, $\mathbf{q}_1$ and $\mathbf{q}_2$ related to the two rotations and the unit vector **n** perpendicular to $z_3$ and **OP** are determined as follows.

$$ \mathbf{n} = \begin{bmatrix} -\sin\phi & \cos\phi & 0 \end{bmatrix}^T $$

$$ \mathbf{q}_1 = \begin{bmatrix} \cos\frac{\pi}{4} & -\sin\frac{\pi}{4}\sin\phi & \sin\frac{\pi}{4}\cos\phi & 0 \end{bmatrix}^T $$

$$ \mathbf{q}_2 = \begin{bmatrix} \cos\frac{\alpha}{2} & 0 & 0 & \sin\frac{\alpha}{2} \end{bmatrix}^T $$

Therefore, the quaternion $\mathbf{q}$ of the coordinate transformation operator (1) is determined by multiplying $\mathbf{q}_2$ and $\mathbf{q}_1$ as in Box I. Substituting $\mathbf{q}$ in the system dynamics (15) and after simplification, the direction of the linear velocity of the spherical shell in this configuration is determined as follows.

$$ \frac{\dot{y}}{\dot{x}} = \frac{u_2\sin\alpha - u_1\cos\alpha + u_1\cos(\alpha+2\phi) + u_2\sin(\alpha+2\phi)}{u_2\cos\alpha + u_1\sin\alpha + u_2\cos(\alpha+2\phi) - u_1\sin(\alpha+2\phi)} = \tan(\alpha+\phi) $$

It is clear that for every input vector, the slope of the robot's trajectory on the ground is equal to $\tan(\alpha+\phi)$ and it is independent of the control inputs. As the physical interpretation of this result, when the robot is in the uncontrollable region, the spherical shell could merely roll along the meridian through the contact point and perpendicular to the great circle $\Gamma$. This is the geometrical meaning of Theorem 1 which has not been presented in the previous studies. ■

**Remark 1.** Let $\mathbf{G}_{1:2}(\mathbf{x})$ be the first two rows of the matrix function $\mathbf{G}$ from the system dynamics (15). By substituting (16) and (17) in $\mathbf{G}_{1:2}(\mathbf{x})$, it is easy to check that the determinant of $\mathbf{G}_{1:2}(\mathbf{x})$ is zero. Consequently, there exists a non-zero control input that leads to zero linear velocity for the spherical shell and causes to its pure spin around the vertical axis. Therefore, when the contact point between the spherical shell and the ground is in the region defined by (16) and (17), the spherical shell could either rolls along the trajectory perpendicular to the great circle $\Gamma$ or spin in place around the vertical axis. More precisely, when the spherical robot is actuated by two internal rotors, it could not rolls along the circle in the plane of the internal rotors.

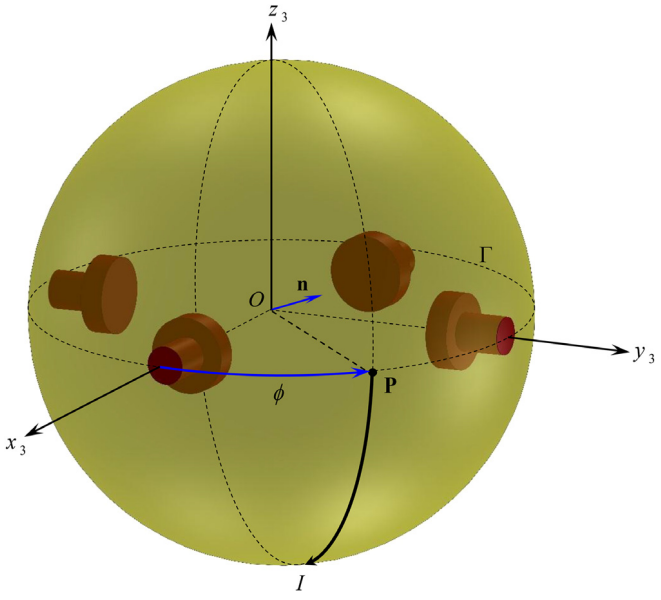


**Fig. 2.** Spherical robot and the uncontrollable region.

**Remark 2.** Although the accessibility distribution is not full rank on the region defined by (16) and (17), a realizable trajectory could be found between every two points in the state space. In other words, when the robot is in a controllable configuration, the accessibility distribution is full rank and all points in the state space are accessible. On the other hand, according to Remark 1, when the robot is in an uncontrollable configuration, there exists a nonzero control input that transmits the system's state to the controllable region in which the accessibility distribution is full rank and all points of the state space are accessible from it.

### 3.2. Model predictive control strategy

Using model predictive control (MPC) technique, a control system of the position and orientation of the spherical robot described in the preceding sections is designed. In the MPC method, the control efforts at each time instant are calculated based on the predicted behavior of the system over a finite prediction horizon and by solving online an open loop optimal control problem [42]. This construction makes it possible to introduce hard constraints on inputs and state variables in control system. Besides, the MPC is a suitable method to control the systems in which the precomputation of feedback law is not feasible [43]. Therefore, the interest of using MPC in trajectory planning and control of robotic

$$ \mathbf{q} = \mathbf{q}_2 \otimes \mathbf{q}_1 = \begin{bmatrix} \frac{\sqrt{2}}{2}\cos\left(\frac{\alpha}{2}\right) & \frac{\sqrt{2}}{2}\sin\left(\frac{\alpha}{2}+\phi\right) & \frac{\sqrt{2}}{2}\cos\left(\frac{\alpha}{2}+\phi\right) & \frac{\sqrt{2}}{2}\sin\left(\frac{\alpha}{2}\right) \end{bmatrix}^T $$

**Box I.**

systems and particularly nonholonomic mobile robots has been increased in recent researches [44–51].

To design a NMPC for the spherical robot, the system dynamics (15) is discretized in the following form.

$$\mathbf{x}^{+} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \int_{t_k}^{t_k+T} \mathbf{G}(\mathbf{x}(\tau))\ \mathbf{u}(\tau)\ d\tau + \mathbf{x} \tag{20}$$

where $\mathbf{x}^{+}$ stands for the successor state, $T$ is the sampling period and $t_k = kT : k = 0, 1, 2, \ldots$ represents the discrete time instants. To determine the integral of the right hand side of (20), the control input vector $\mathbf{u}(t)$ should be discretized during the robot's motion. However, the control inputs are defined as the rotors' angular velocity and it is not realistic to be discontinuous functions of time. Therefore, $\mathbf{u}(t)$ is assumed to be continuous and piecewise linear in each sampling period. In other words, the control inputs are discretized using the first-order hold (FOH) sampling as follows:

$$\begin{aligned} &\mathbf{u}(k=0) = \mathbf{0} \\ &\mathbf{u}_k(t) = \boldsymbol{FOH}_k(t) = (t - t_k)\,\mathbf{h} + \mathbf{u}(k)\,; \quad t_k \le t \le t_{k+1} \end{aligned} \tag{21}$$

Where $\mathbf{u}(k)$ is the value of the control input vector at time $t_k$, $\mathbf{u}_k(t)$ denotes the values of the control input vector in the interval, $[t_k, t_{k+1}]$ and $\mathbf{h}$ is the vector of input changes per unit time defined as:

$$\mathbf{h} = \frac{\mathbf{u}(k+1) - \mathbf{u}(k)}{t_{k+1} - t_k} \tag{22}$$

Using (21) and (22), the successor state is determined from (20) and the trajectory of the system is calculated using the following difference equation iteratively.

$$\begin{aligned} &\mathbf{x}(\mathbf{x}_0, k+1) = \mathbf{f}(\mathbf{x}(\mathbf{x}_0, k), \mathbf{u}(k+1))\,; \quad \mathbf{x}(\mathbf{x}_0, 0) = \mathbf{x}_0 \\ &\mathbf{u}_k(t) = \boldsymbol{FOH}_k(t)\,; \quad t_k \le t \le t_{k+1} \\ &\mathbf{u}(k=0) = \mathbf{0} \\ &\mathbf{x} \in \mathbf{X} \subset \mathbb{R}^6; \quad \mathbf{u}(k) \in \mathbf{U} \subset \mathbb{R}^2 \end{aligned} \tag{23}$$

where $\mathbf{x}(\mathbf{x}_0, k)$ denotes the state variables vector at time, $t_k = kT$ initialized with $\mathbf{x}_0$; $\mathbf{X}$ and $\mathbf{U}$ denote the state and input constraint sets, respectively. Since, there exists no state constraint on the spherical robot dynamics (15), $\mathbf{X} = \mathbb{R}^2 \times \mathbb{S}^4_1$. Furthermore, the input constraint set associated with the maximum allowable actuators' input, is defined as $\boldsymbol{U} = \left\{ \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T \in \mathbb{R}^2,\ -u_{\max} \le u_i \le u_{\max},\ i = 1, 2 \right\}$.

**Remark 3.** Using the FOH sampling, the control input in each sampling interval, $[t_k, t_{k+1}]$ changes linearly from its value at the end of the previous sampling period, $\mathbf{u}(k)$ to its final value at the end of the current sampling interval, $\mathbf{u}(k+1)$ . This shape of control input leads to continuous piecewise linear angular velocity and piecewise constant angular acceleration of the rotors which in turn requires piecewise constant torques by the motors.

To design a NMPC for regulation of the discrete-time system (23), the following cost function is defined over a finite prediction horizon of length, $N$.

$$\begin{aligned} &V_N(\mathbf{x}(\mathbf{x}_0, k) - \mathbf{x}_d, \mathbf{u}(.)) \\ &\quad = \sum_{m=0}^{N-1} l(\mathbf{x}(\mathbf{x}_0, k+m) - \mathbf{x}_d, \mathbf{u}(k+m+1)) \\ &\quad + V_f(\mathbf{x}(\mathbf{x}_0, k+N) - \mathbf{x}_d) \end{aligned} \tag{24}$$

where $l(\cdot, \cdot)$ is a positive definite stage cost, $V_f(\cdot)$ is a positive definite terminal cost, $\mathbf{x}_d$ is the desired state and $V_N(0, 0) = 0$. Therefore, the problem of stabilization of the spherical robot (15) is formulated as the following optimal control problem, $\mathbf{OCP}_N$ to compute the optimal control sequence, $\tilde{\mathbf{u}}_N^* \in \mathbf{U}^N$.

$$\begin{aligned} &\mathbf{OCP}_N : \min_{\tilde{\mathbf{u}} \sim (\cdot) \in \mathbf{U}^N} \left\{ \begin{array}{l} V_N(\mathbf{x}(\mathbf{x}_0, k) - \mathbf{x}_d, \mathbf{u}) \\ = \sum_{m=0}^{N-1} l(\mathbf{x}(\mathbf{x}_0, k+m) - \mathbf{x}_d, \mathbf{u}(k+m+1)) \\ + V_f(\mathbf{x}(\mathbf{x}_0, k+N) - \mathbf{x}_d) \end{array} \right\} \\ &S.t : \\ &\quad \mathbf{A} : \mathbf{x}(\mathbf{x}_0, k+m) = \mathbf{f}(\mathbf{x}(\mathbf{x}_0, k+m-1), \mathbf{u}(k+m)) \\ &\quad \mathbf{B} : \mathbf{x}(\mathbf{x}_0, 0) = \mathbf{x}_0 \\ &\quad \mathbf{C} : \mathbf{x}(\mathbf{x}_0, k+m) \in \mathbf{X} \\ &\quad \mathbf{D} : \mathbf{x}(\mathbf{x}_0, k+N) \in \mathbf{X}_f \subseteq \mathbf{X} \\ &\quad \mathbf{E} : \mathbf{u}(k=0) = \mathbf{0} \\ &\quad \mathbf{F} : \mathbf{u}_k(t) = \boldsymbol{FOH}_k(t)\,; \quad t_k \le t \le t_{k+1} \end{aligned} \tag{25}$$

where, $\mathbf{X}_f$ is a terminal constraint set. The result of the above $\mathbf{OCP}_N$ is a sequence of $N$ input vectors as:

$$\tilde{\mathbf{u}}_N^* = \left\{ \mathbf{u}^*(k+1), \mathbf{u}^*(k+2), \ldots, \mathbf{u}^*(k+N) \right\} \in \mathbf{U}^N \tag{26}$$

which leads to the optimal value function $V_N^*$ and yields the following optimal states sequence.

$$\tilde{\mathbf{x}}_N^* = \left\{ \mathbf{x}^*(k+1), \mathbf{x}^*(k+2), \ldots, \mathbf{x}^*(k+N) \right\} \in \mathbf{X}^N \tag{27}$$

The control feedback law at time, $t_k$ is defined as the first element of the control sequence (26) as follows.

$$\begin{aligned} &\mathbf{u}(k+1) = \boldsymbol{\mu}(\mathbf{x}(\mathbf{x}_0, k)) = \mathbf{u}^*(k+1)\,; \\ &\mathbf{u}_k(t) = \boldsymbol{FOH}_k(t)\,; \quad t_k \le t \le t_{k+1} \end{aligned} \tag{28}$$

Which leads to the following closed loop dynamics

$$\begin{aligned} &\mathbf{x}(\mathbf{x}_0, k+1) = \mathbf{f}(\mathbf{x}(\mathbf{x}_0, k), \boldsymbol{\mu}(\mathbf{x}(\mathbf{x}_0, k)))\,; \\ &\mathbf{x}(\mathbf{x}_0, 0) = \mathbf{x}_0; \quad \mathbf{u}_k(t) = \boldsymbol{FOH}_k(t) \end{aligned} \tag{29}$$

The above-explained NMPC strategy is summarized in the following seven steps.

***Algorithm 1: NMPC***

***Step 1:*** initialize the control algorithm by initial guess for the control sequence $\tilde{\mathbf{u}}$.

***Step 2:*** compute the system's state $\mathbf{x}(\mathbf{x}_0, k)$ at time $t_k$.

***Step 3:*** solve the optimal control problem (25) and find the optimal control sequence $\tilde{\mathbf{u}}^*$.

***Step 4***: set the first element of the optimal control sequence as the feedback control input at time $t_k$ (Eq. (28)).

***Step 5:*** exert the feedback control input to the system dynamics and determine the next state by Eq. (29).

***Step 6:*** add a zero element at the end of the remaining elements of the optimal control sequence, and set the resulting sequence as an initial guess for the next step $\tilde{\mathbf{u}} = \{\mathbf{u}^*(k+2), \ldots, \mathbf{u}^*(k+N), \mathbf{0}\}$.

***Step 7:*** set $k \to k+1$ and go to step 2.

The design parameters of the feedback control system $N$, $l(\cdot, \cdot)$, $V_f(\cdot)$ and $\mathbf{X}_f$ are determined in such a way that the stability of the closed loop system (29) is guaranteed.

### 3.3. Stability analysis

Although, it is shown that MPC with infinite prediction horizon is generally stable, the stability of MPC within a finite prediction horizon is not very straightforward. On the other hand, the implementation of MPC with infinite prediction horizon is not possible in real world applications due to high computational burden. The conditions to guarantee the stability of MPC within a finite prediction horizon have been investigated in several studies mostly based on a value function like Lyapunov function candidate. A similar method is considered here to analyze the stability of the closed

loop system (29) and determine the ingredients of Algorithm 1. Dealing with the main stability proof, some required definitions and assumptions are presented.

**Definition 1.** The set $\mathbf{X}_f \subset \mathbf{X}$ is called forward invariant for the system dynamics (20) if for each $\mathbf{x} \in \mathbf{X}_f$ there exists a control input $\mathbf{u} \in \mathbf{U}$ such that $\mathbf{x}^+ = \mathbf{f}(\mathbf{x}, \mathbf{u}) \in \mathbf{X}_f$.

**Definition 2.** The terminal cost $V_f(\cdot)$ is called a local control Lyapunov function (CLF) on $\mathbf{X}_f$ if for each $\mathbf{x} \in \mathbf{X}_f$ there exists a control input, $\mathbf{u} \in \mathbf{U}$such that:

$$\mathbf{x}^+ = \mathbf{f}(\mathbf{x}, \mathbf{u}) \in \mathbf{X}_f \tag{30}$$

$$V_f(\mathbf{f}(\mathbf{x}, \mathbf{u})) - V_f(\mathbf{x}) + l(\mathbf{x}, \mathbf{u}) \leq 0 \tag{31}$$

The terminal cost $V_f(\cdot)$ is called a global CLF if $\mathbf{X}_f$ includes the whole state space.

**Definition 3.** Two classes of comparison functions are defined as follows.

$$\begin{aligned}\kappa &= \{\eta(\cdot) : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \mid \eta(0) = 0,\\ &\quad \eta \text{ is continuous \& strictly increasing}\}\\ \kappa_\infty &= \{\eta(\cdot) : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \mid \eta \in \kappa,\ \eta \text{ is unbounded}\}\end{aligned} \tag{32}$$

Although finding a local CLF is simpler than a global one, it is very desirable to solve the $\mathbf{OCP}_N$ without state constraints. Therefore, the following assumptions are considered.

**Assumption 1.** Suppose that the terminal set $\mathbf{X}_f$ covers the whole state space and therefore, it is a global forward invariant set. Mathematically:

$$\mathbf{X}_f = \mathbb{R}^2 \times \mathbb{S}_1^4 \quad \Rightarrow \quad \forall \mathbf{x} \in \mathbf{X}_f \text{ and } \mathbf{u} \in \mathbf{U},\ \mathbf{x}^+ = \mathbf{f}(\mathbf{x}, \mathbf{u}) \in \mathbf{X}_f$$

**Assumption 2.** The stage cost includes two quadratic terms of state variables and control inputs and the terminal cost is a quadratic function of the state variables as:

$$\begin{aligned}l(\mathbf{x}, \mathbf{u}) &= \mathbf{x}^T \mathbf{Q}_l \mathbf{x} + \mathbf{u}^T \mathbf{Q}_u \mathbf{u}\\ V_f(\mathbf{x}) &= \mathbf{x}^T \mathbf{Q}_f \mathbf{x}\end{aligned} \tag{33}$$

where, both $\mathbf{Q}_l$ and $\mathbf{Q}_f$ are 6-dimensional positive definite matrices and $\mathbf{Q}_u$ is a 2-dimensional one.

**Assumption 3.** It is assumed that for every pair $(\mathbf{x}, \mathbf{x}_d)$, there exists a control input $\mathbf{u} \in \mathbf{U}$ to move the states of the system (23) to a new state $\mathbf{x}^+ \in \mathbb{R}^2 \times \mathbb{S}_1^4$ which is closer to $\mathbf{x}_d$. More precisely:

$$\Delta = \left\|\mathbf{x}^+ - \mathbf{x}_d\right\|^2 - \|\mathbf{x} - \mathbf{x}_d\|^2 < 0$$

**Remark 4.** The logic behind this assumption refers to the controllability properties of the spherical robot (Theorem 1 and remarks therein). Indeed, as explained in Remark 2, when the robot is in a controllable configuration, every state of the system is accessible from another and therefore, there exists a trajectory from the current state to a new state closer to the desired configuration. On the other hand, in an uncontrollable configuration, the only restriction on the robot's trajectory is orthogonality of the locus of the contact points on the sphere to the circle of the internal rotors at their intersections. Therefore, several trajectories could be found to satisfy this restriction and transmit the system's state to a new configuration closer to the desired state. It is to be noted that, this assumption is the result of the controllability analyses and merely states the existence of a control input that moves the robot to a new configuration closer to desired state. However, the monotonically decreasing of the cost function (24) and step by step convergence of the system's state variables to the desired configuration in control algorithm 1 will be proven by Theorem 3.

**Theorem 2.** *Suppose that Assumption 1 to 3 are satisfied, $\mathbf{Q}_l$ is an identity matrix, $\mathbf{Q}_u$ is a diagonal matrix with equal diagonal elements and $\mathbf{Q}_f$ is a diagonal matrix with equal and sufficiently large diagonal elements, then the terminal cost function $V_f(\mathbf{x})$ is a CLF on $\mathbb{R}^2 \times \mathbb{S}_1^4 - \mathbf{D}$ where $\mathbf{D}$ is a small neighborhood of the desired state.*

**Proof.** Regarding the diagonal matrices as, $\mathbf{Q}_f = diag(\beta, \beta, \beta, \beta, \beta, \beta)$ and $\mathbf{Q}_u = diag(\alpha, \alpha)$, where $\alpha$ and $\beta$ are strictly positive constants, the stage and the terminal costs could be rewritten as:

$$\begin{aligned}l(\mathbf{x}, \mathbf{u}) &= \|\mathbf{x} - \mathbf{x}_d\|^2 + \alpha \|\mathbf{u}\|^2\\ V_f(\mathbf{x}) &= \beta \|\mathbf{x} - \mathbf{x}_d\|^2\end{aligned} \tag{34}$$

Substituting (34) in (31) yields:

$$\begin{aligned}&\beta\left(\left\|\mathbf{x}^+ - \mathbf{x}_d\right\|^2\right) - \beta\left(\|\mathbf{x} - \mathbf{x}_d\|^2\right) + \left(\|\mathbf{x} - \mathbf{x}_d\|^2\right)\\ &\quad + \alpha\left(\|\mathbf{u}\|^2\right) \leq 0\end{aligned} \tag{35}$$

Considering Assumption 3 and the definition of the control input constraint set, $\mathbf{U}$ gives:

$$\begin{aligned}&\beta \cdot \Delta + \left(\|\mathbf{x} - \mathbf{x}_d\|^2\right) + \alpha\left(\|\mathbf{u}\|^2\right) \leq \beta \cdot \Delta + \left(\|\mathbf{x} - \mathbf{x}_d\|^2\right)\\ &\quad + 2\alpha u_{\max}^2 \leq 0\end{aligned} \tag{36}$$

Therefore, by choosing $\beta \geq \frac{\left(\|\mathbf{x}-\mathbf{x}_d\|^2 + 2\alpha u_{\max}^2\right)}{|\Delta|}$ the inequality (36) is satisfied and the terminal cost $V_f(\mathbf{x})$ is a CLF. However, precomputation of $\mathbf{x}^+$ and $\Delta$ in each time step is not possible. Consequently, $\beta$ should be chosen sufficiently large to satisfy inequality (36) over the whole of the state space except a small neighborhood of the desired state. ■

**Remark 5.** In each system's state, $\Delta$ satisfies $\Delta \leq \|\mathbf{x} - \mathbf{x}_d\|^2$ and its value depends on of $\mathbf{x}$, $u_{\max}$ and the robot's physical parameters. In a small vicinity of the desired state, both $\Delta$ and $\|\mathbf{x} - \mathbf{x}_d\|^2$ are small, $\Delta \to 0$ *as* $\mathbf{x} \to \mathbf{x}_d$; and by use of a fixed, $\beta$ the inequality (36) may not be satisfied providing $|\Delta| < \frac{\left(\|\mathbf{x}-\mathbf{x}_d\|^2 + 2\alpha u_{\max}^2\right)}{\beta}$. Therefore, using a sufficiently large value for $\beta$ guarantees that the vicinity is a small neighborhood of the desired state. However, the value of $\beta$ and $\alpha$ could be tuned by trial and error to achieve acceptable results.

**Assumption 4.** There exist $\delta_1(\cdot), \delta_2(\cdot) \in \kappa_\infty$ such that for every $\mathbf{x} \in \mathbf{X}$ and $\mathbf{u} \in \mathbf{U}$ we have:

$$V_f(\mathbf{x} - \mathbf{x}_d) \leq \delta_1(\|\mathbf{x} - \mathbf{x}_d\|) \quad \text{and} \quad l(\mathbf{x} - \mathbf{x}_d, u) \geq \delta_2(\|\mathbf{x} - \mathbf{x}_d\|).$$

It is to be noted that since stage and terminal costs are assumed to be quadratic functions of $\mathbf{x} - \mathbf{x}_d$ and $\mathbf{u}$ (Assumption 2) and $\|\mathbf{u}\|^2 \leq 2u_{\max}^2$, the terms $\delta_1$ and $\delta_2$ in Assumption 4 could be easily specified

**Theorem 3.** *Suppose that Assumptions 1 to 4 are satisfied and $\mathbf{Q}_l$, $\mathbf{Q}_f$ and $\mathbf{Q}_u$ are chosen as in Theorem 2, then the closed loop system (29) is asymptotically stable to a neighborhood of the desired state ($\mathbf{D}$ : $\|\mathbf{x} - \mathbf{x}_d\| < \varepsilon$) explained in Remark 5.*

**Proof.** Considering the value function $V_N^*(\mathbf{x}(\mathbf{x}_0, k))$ as a Lyapunov function candidate the definition of the cost function $V_N$ by Eq. (24) yields:

$$\begin{aligned}V_N(\mathbf{x}(\mathbf{x}_0, k), \mathbf{x}_d, \mathbf{u}) &= \sum_{m=0}^{N-1} l(\mathbf{x}(\mathbf{x}_0, k+m) - \mathbf{x}_d, \mathbf{u}(k+m+1))\\ &\quad + V_f(\mathbf{x}(\mathbf{x}_0, k+N) - \mathbf{x}_d)\\ &= \sum_{m=0}^{N-2} l(\mathbf{x}(\mathbf{x}_0, k+m) - \mathbf{x}_d, \mathbf{u}(k+m+1))\\ &\quad + V_f(\mathbf{x}(\mathbf{x}_0, k+N) - \mathbf{x}_d)\\ &\quad + l(\mathbf{x}(\mathbf{x}_0, k+N-1) - \mathbf{x}_d, \mathbf{u}(k+N))\end{aligned} \tag{37}$$

On the other hand $V_{N-1}(\mathbf{x}(\mathbf{x}_0,k),\mathbf{x}_d,\mathbf{u})$ could be written as follows.

$$V_{N-1}(\mathbf{x}(\mathbf{x}_0,k),\mathbf{x}_d,\mathbf{u}) = \sum_{m=0}^{N-2} l(\mathbf{x}(\mathbf{x}_0,k+m)-\mathbf{x}_d,\mathbf{u}(k+m+1)) + V_f(\mathbf{x}(\mathbf{x}_0,k+N-1)-\mathbf{x}_d)$$

Using $V_{N-1}(\mathbf{x}(\mathbf{x}_0,k),\mathbf{x}_d,\mathbf{u})$ in (37) yields:

$$\begin{aligned} &V_N(\mathbf{x}(\mathbf{x}_0,k),\mathbf{x}_d,\mathbf{u}) - V_{N-1}(\mathbf{x}(\mathbf{x}_0,k),\mathbf{x}_d,\mathbf{u}) \\ &\quad = V_f(\mathbf{x}(\mathbf{x}_0,k+N)-\mathbf{x}_d) \\ &\quad -V_f(\mathbf{x}(\mathbf{x}_0,k+N-1)-\mathbf{x}_d) \\ &\quad +l(\mathbf{x}(\mathbf{x}_0,k+N-1)-\mathbf{x}_d,\mathbf{u}(k+N)) \end{aligned} \tag{38}$$

Applying the result of Theorem 2 in the right-hand side of (38) leads to:

$$V_N(\mathbf{x}(\mathbf{x}_0,k),\mathbf{x}_d,\mathbf{u}) \le V_{N-1}(\mathbf{x}(\mathbf{x}_0,k),\mathbf{x}_d,\mathbf{u})$$

Therefore, for the value function $V_N^*(\mathbf{x}(\mathbf{x}_0,k))$ we have:

$$\begin{aligned} V_N^*(\mathbf{x}(\mathbf{x}_0,k)) &= \inf_{\tilde{\mathbf{u}}(\cdot)\in\mathbf{U}^N} V_N(\mathbf{x}(\mathbf{x}_0,k),\mathbf{x}_d,\mathbf{u}) \\ &\le \inf_{\tilde{\mathbf{u}}(\cdot)\in\mathbf{U}^{N-1}} V_{N-1}(\mathbf{x}(\mathbf{x}_0,k),\mathbf{x}_d,\mathbf{u}) \\ &= V_{N-1}^*(\mathbf{x}(\mathbf{x}_0,k)) \end{aligned}$$

$$V_N^*(\mathbf{x}(\mathbf{x}_0,k)) \le V_{N-1}^*(\mathbf{x}(\mathbf{x}_0,k)) \tag{39}$$

Now, using dynamic programming principle [52] the following relation is achieved.

$$V_N^*\left(\mathbf{x}^*(\mathbf{x}_0,k)\right) = V_{N-1}^*\left(\mathbf{x}^*(\mathbf{x}_0,k+1)\right) + l\left(\mathbf{x}^*(\mathbf{x}_0,k)-\mathbf{x}_d,\mathbf{u}^*(k+1)\right)$$

And therefore,

$$V_N^*(\mathbf{x}(\mathbf{x}_0,k)) \ge l\left(\mathbf{x}(\mathbf{x}_0,k)-\mathbf{x}_d,\mathbf{u}^*(k+1)\right) \ge \delta_2(\|\mathbf{x}-\mathbf{x}_d\|) \tag{40}$$

On the other hand, using (39) consecutively from $N$ to 1 leads to:

$$\begin{aligned} &V_N^*(\mathbf{x}(\mathbf{x}_0,k)) \le V_{N-1}^*(\mathbf{x}(\mathbf{x}_0,k)) \le \cdots \le V_1^*(\mathbf{x}(\mathbf{x}_0,k)) \\ &\le V_f(\mathbf{x}(\mathbf{x}_0,k+1)) \le \delta_1(\|\mathbf{x}(\mathbf{x}_0,k+1)-\mathbf{x}_d\|) \end{aligned} \tag{41}$$

Furthermore, using the result of Theorem 2, it could be concluded that there exists a control input $\mathbf{u}(k+1)$ that the following is satisfied [53]

$$\begin{aligned} &V_N^*(\mathbf{f}(\mathbf{x}(\mathbf{x}_0,k),\mathbf{u}(k+1))) - V_N^*(\mathbf{x}(\mathbf{x}_0,k)) \\ &\le -l(\mathbf{x}(\mathbf{x}_0,k)-\mathbf{x}_d,\mathbf{u}(k+1)) \le -\delta_2(\|\mathbf{x}-\mathbf{x}_d\|) \end{aligned} \tag{42}$$

The conjugation of (40), (41) and (42) implies that, $V_N^*(\mathbf{x}(\mathbf{x}_0,k))$ is a Lyapunov function and the closed loop system (29) is asymptotically stable to the desired state. Furthermore, when the trajectory of the system reaches the neighborhood, $\mathbf{D}: \|\mathbf{x}-\mathbf{x}_d\| < \varepsilon$ the terminal cost function $V_f(\cdot)$ may not be a CLF on this set (Theorem 2) and therefore, the above results are not held on this region. However, suppose that the state vector of the system at time $t_k$ is $\mathbf{x}(\mathbf{x}_0,k) \in \mathbf{D}$, then $\tilde{\mathbf{u}}_N = \{\mathbf{0},\mathbf{0},\ldots,\mathbf{0}\}$ could be regarded as an initial guess for the control input sequence in $\mathbf{OCP}_N$ for the next step. This sequence leads to $V_N\left(\mathbf{x}^*(k+1),\mathbf{x}_d,\tilde{\mathbf{u}}_N\right) \le (N-1)\varepsilon + \beta\varepsilon$ which is an upper bound for $V_N^*(\mathbf{x}(\mathbf{x}_0,k+1))$. Therefore, choosing $\beta \gg N-1$ implies that the trajectory of the system remains in $\mathbf{D}$ for $t \ge t_k$ or if the trajectory exits from $\mathbf{D}$, it returns back to a state in $\mathbf{D}$ in the future which is closer than $\mathbf{x}(\mathbf{x}_0,k) \in \mathbf{D}$ to the desired state. ■

## 4. Results and discussion

In this section, the designed NMPC (Algorithm 1) is implemented in MATLAB software and its performance is assessed by computer simulations. The actuators' input are calculated by the control system at each sampling time to move the spherical robot to the desired position and orientation. The robot's physical parameters are assumed according to Table 1.

In the designed NMPC (Algorithm 1), the optimal control problem $\mathbf{OCP}_N$ (25) must be solved in each sampling time. Therefore, the resulting constrained nonlinear optimization problem is solved using the well-known sequential quadratic programming approach (SQP) to calculate the control input sequence (26). In addition, since introducing the equality constraints of the system dynamics (25.A) in the optimization problem increases the computational time, these constraints are considered as a part of the cost function (24) (recursive discretization method) [52]. Consequently, the resulting optimization problem is solved by only considering the constraints on the actuators' input bounds defined by the input constraint set, $\mathbf{U}$. Choosing $\beta = 10^8$, $\alpha = \frac{1}{2}$, the prediction horizon length, $N = 4$ the sampling period length, $T = .5$ s the maximum allowable actuators input $u_{\max} = 100$ rad/s and using fmincon MATLAB function for solving $\mathbf{OCP}_N$ by SQP approach, the performance of the designed NMPC is evaluated in the following four cases.

***Case 1: Stabilization to origin from initial position and attitude off-tracks***

It is assumed that the robot is initially at rest and starts its motion from the initial state $\mathbf{x}(\mathbf{x}_0,0) = \mathbf{x}_0 = \begin{bmatrix}2 & -3 & .03 & 0.454 & 0.88 & 0.1363\end{bmatrix}^T$. Our goal is to move the robot to the desired state $\mathbf{x}_d = \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}^T$ using the feedback control law (28). The effectiveness of the control system is assessed by simulation and the results are represented in Fig. 3.

***Case 2: Stabilization to uncontrollable region from an uncontrollable configuration***

It is assumed that the robot is initially at rest in an uncontrollable state $\mathbf{x}(\mathbf{x}_0,0) = \mathbf{x}_0 = \begin{bmatrix}1 & 2 & .5 & .5 & .5 & .5\end{bmatrix}^T$ and should move to another uncontrollable configuration at the origin of frame $\{1\}\,\mathbf{x}_d = \begin{bmatrix}0 & 0 & 0.7071 & 0 & 0.7071 & 0\end{bmatrix}^T$. Using the designed NMPC, the simulation results are illustrated in Fig. 4.

***Case 3: Stabilization to origin from zero error on position and initial attitude off-track***

It is assumed that the robot is initially at rest in the initial state $\mathbf{x}(\mathbf{x}_0,0) = \mathbf{x}_0 = \begin{bmatrix}0 & 0 & .6521 & .1132 & .1115 & .7413\end{bmatrix}^T$ at the origin of frame {1} and should reach the desired state at the same position with different orientation as, $\mathbf{x}_d = \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}^T$. Fig. 5 shows the simulation results for the robot's trajectory, position of the spherical shell, values of Euler's parameters and actuators inputs.

***Case 4: Stabilization to uncontrollable region in the presence of bounded disturbance and parameters uncertainty***

The robot is initially at rest in an uncontrollable configuration at the origin of frame $\{1\}\,\mathbf{x}(\mathbf{x}_0,0) = \mathbf{x}_0 = \begin{bmatrix}0 & 0 & .2734 & .5628 & .4281 & .6521\end{bmatrix}^T$ and should reach another point at the origin of frame {1} in the uncontrollable region $\mathbf{x}_d = \begin{bmatrix}0 & 0 & .7071 & 0 & .7071 & 0\end{bmatrix}^T$. To evaluate the robustness of the designed control system, the white Gaussian process is considered to affect the robot's model (23) as:

$$\begin{aligned} &\mathbf{x}(\mathbf{x}_0,k+1) = \mathbf{f}(\mathbf{x}(\mathbf{x}_0,k),\mathbf{u}(k+1)) + \mathbf{W}(k); \quad \mathbf{x}(\mathbf{x}_0,0) = \mathbf{x}_0 \\ &\mathbf{x} \in \mathbf{X} \subset \mathbb{R}^6; \quad \mathbf{u} \in \mathbf{U} \subset \mathbb{R}^2 \\ &\mathbf{W}(k) \subset \mathbb{R}^6 \end{aligned} \tag{43}$$

where, $\mathbf{W}(k)$ denotes the additive Gaussian disturbance. Furthermore, the robot's physical parameters are assumed to be subjected

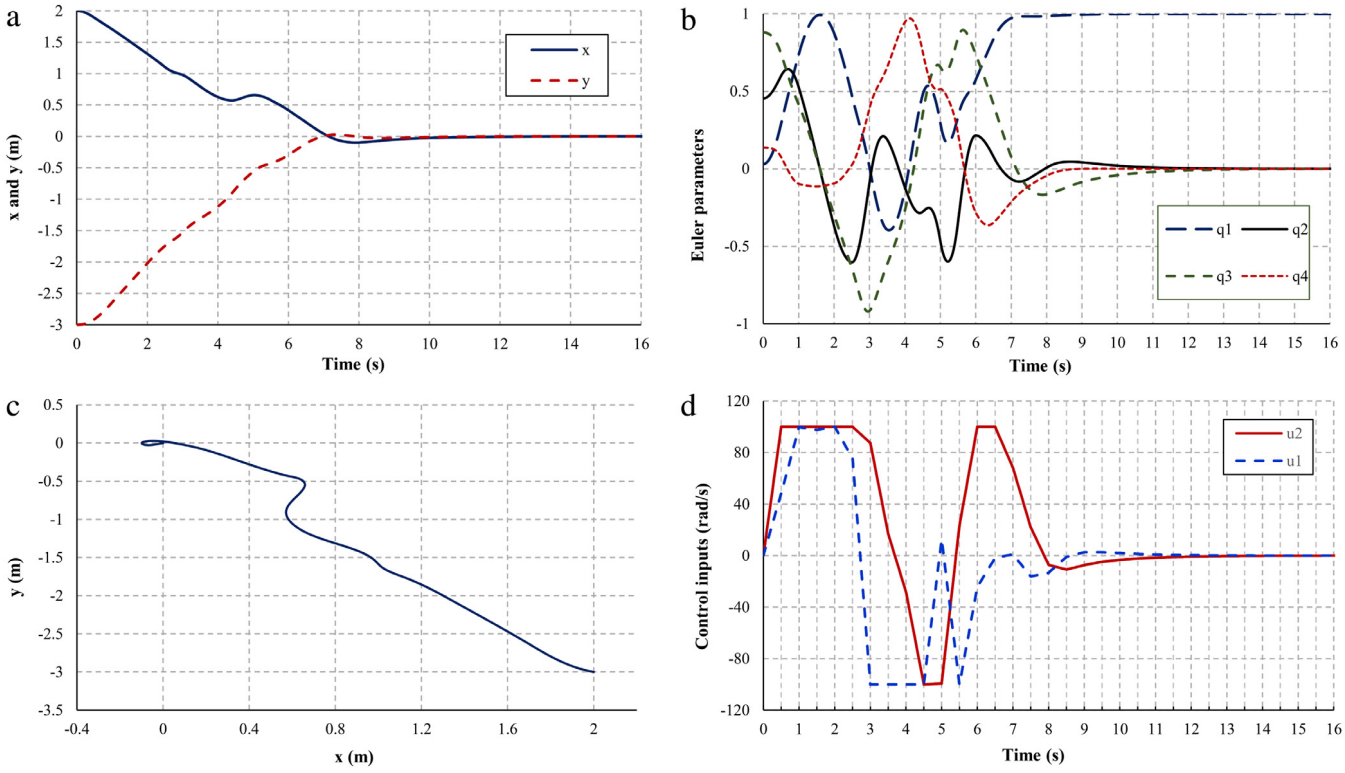


**Fig. 3.** Simulation results for case 1, a: position of the spherical shell, b: Euler's parameters, c: robot's trajectory on the contact plane and d: actuators' input.

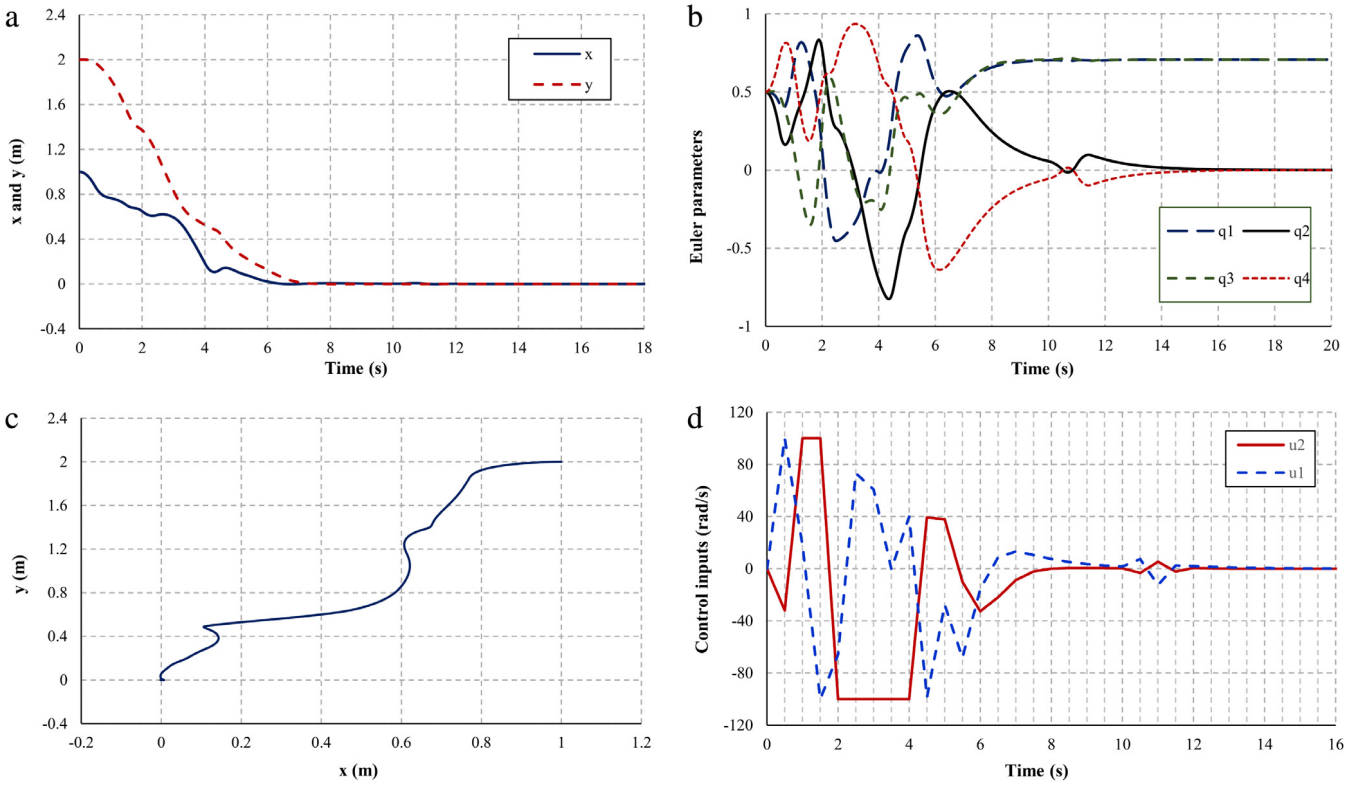


**Fig. 4.** Simulation results for case 2, a: position of the spherical shell, b: Euler's parameters, c: robot's trajectory on the contact plane and d: actuators' input.

**Table 1**
Spherical robot physical parameters.

| Parameter | $M_S$ (kg) | $R_S$ (m) | $I_S$ (kg.m$^2$) | $d$ (m) | $M_{cw}$ (kg) | $M_r$ (kg) | $I_1, I_{c1}$ (kg m$^2$) | $I_2, I_{c2}$ (kg m$^2$) |
|---|---|---|---|---|---|---|---|---|
| Value | 1 | 0.3 | 0.06 | 0.15 | 0.7 | 0.7 | 0.008 | 0.004 |

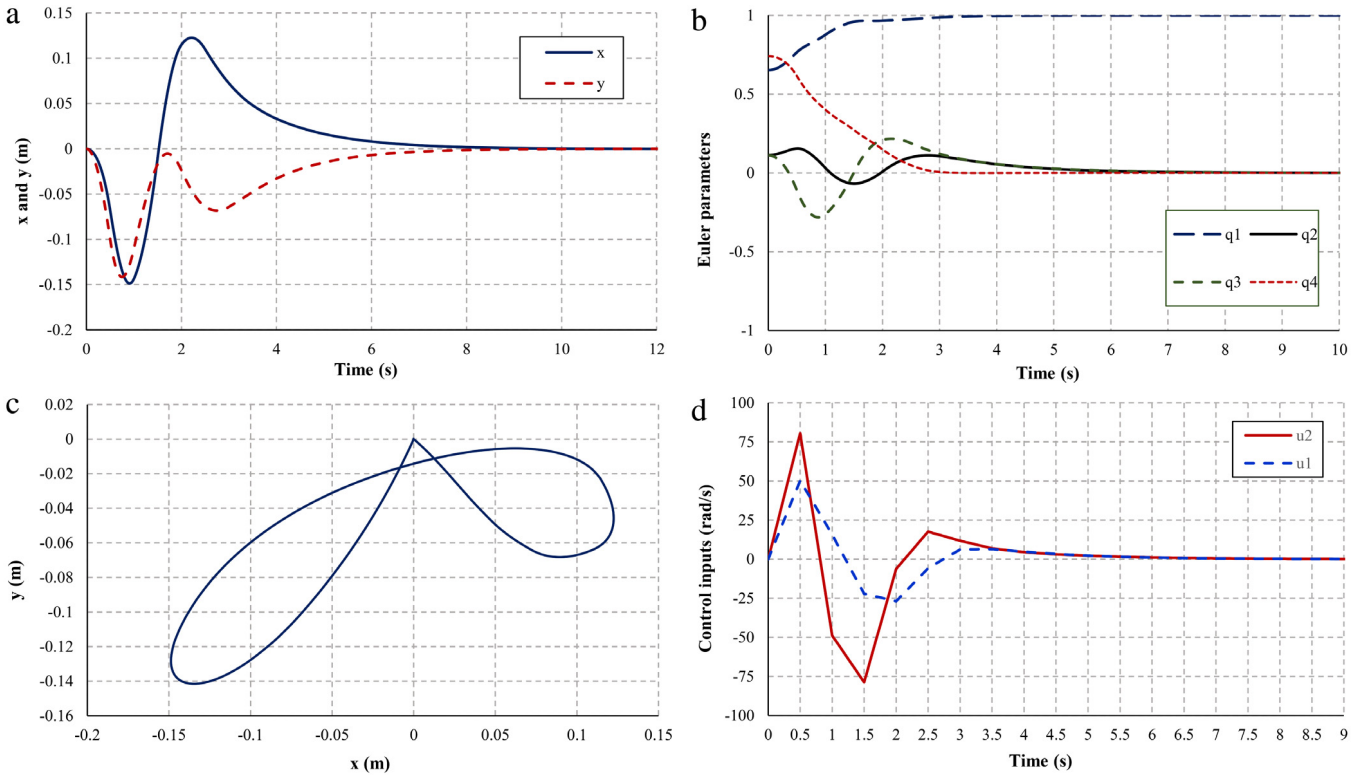


**Fig. 5.** Simulation results for case 3, a: position of the spherical shell, b: Euler's parameters, c: robot's trajectory on the contact plane and d: actuators' input.

up to 3% uncertainty with respect to their nominal values The simulation of the control system results in the robot trajectory, position of the spherical shell, value of Euler's parameters and actuators input of Fig. 6.

According to the obtained results some highlights are listed below:

- The designed control system could stabilize the spherical robot to every desired position and orientation in different situations.
- Using the FOH sampling (Eq. (21)) and the feedback control law (28), the control input at each time instant is a function of time $t$ and the system's state at the end of last sampling interval. Therefore, using the proposed control method, a continuous time varying feedback control law is achieved for the state space model (15) as, $\mathbf{u}_k(t) = \mathbf{u}(\mathbf{x}(\mathbf{x}_0, k), t)$ ; $t_k \leq t \leq t_{k+1}$. This property satisfies Brockett's necessary conditions for stabilization of nonholonomic systems [8].
- As stated in Corollary 1, it is obvious from Fig. 7 that every trajectory of the robot is such that the locus of the contact points on the sphere is perpendicular to the great circle in the plane of the internal rotors at their intersections.
- It is clear from Fig. 7 and the results of case 2 and case 4 that though the robot is not controllable over the whole of the state space, the stabilization to every uncontrollable state could be achieved through the designed control system.

- The nonholonomic constraints imposed by the perfect rolling condition are relations between the position and the orientation of the spherical shell and their time derivatives as represented by (7). Therefore, every change in the orientation leads to change in the position of the spherical shell and vice versa. Consequently, the trajectory of the spherical shell on the contact plane may not be the shortest one. Indeed, the control system may have to increase the error on position or orientation to find a trajectory from the current state to the desired position and orientation in the future (Figs. 3c and 4c).

- The required motor's torque in the above simulations could be estimated using Newton–Euler equation for each rotors as follows.

$$^3\mathbf{M} = \mathbf{I}_r\, ^3\dot{\omega}_r + {}^3\boldsymbol{\omega}_3 \times \mathbf{I}_r\, ^3\boldsymbol{\omega}_r \tag{44}$$

Where $^3\mathbf{M}$ stands for the resultant torque vector applied on the rotors described in {3}. The required torques of motors 1 and 2 along rotation axes, $x_3$ and $y_3$ are equal to the corresponding components of $^3\mathbf{M}$, respectively. Using (5) and (6) in (44), the required motor torques are simplified in the following form:

$$\begin{aligned} M_1 &= I_1\left(^3\alpha_{r1x}\right) = I_1\left(^3\dot{\omega}_{3x} + \dot{u}_1\right) \\ M_2 &= I_1\left(^3\alpha_{r2y}\right) = I_1\left(^3\dot{\omega}_{3y} + \dot{u}_2\right) \end{aligned} \tag{45}$$

where $^3\alpha_{r1x}$ and $^3\alpha_{r2y}$ denote the angular accelerations of the rotors with respect to inertial reference frame along the $x$ and $y$ directions of frame {3} respectively. Using the numerical differentiation in each sampling

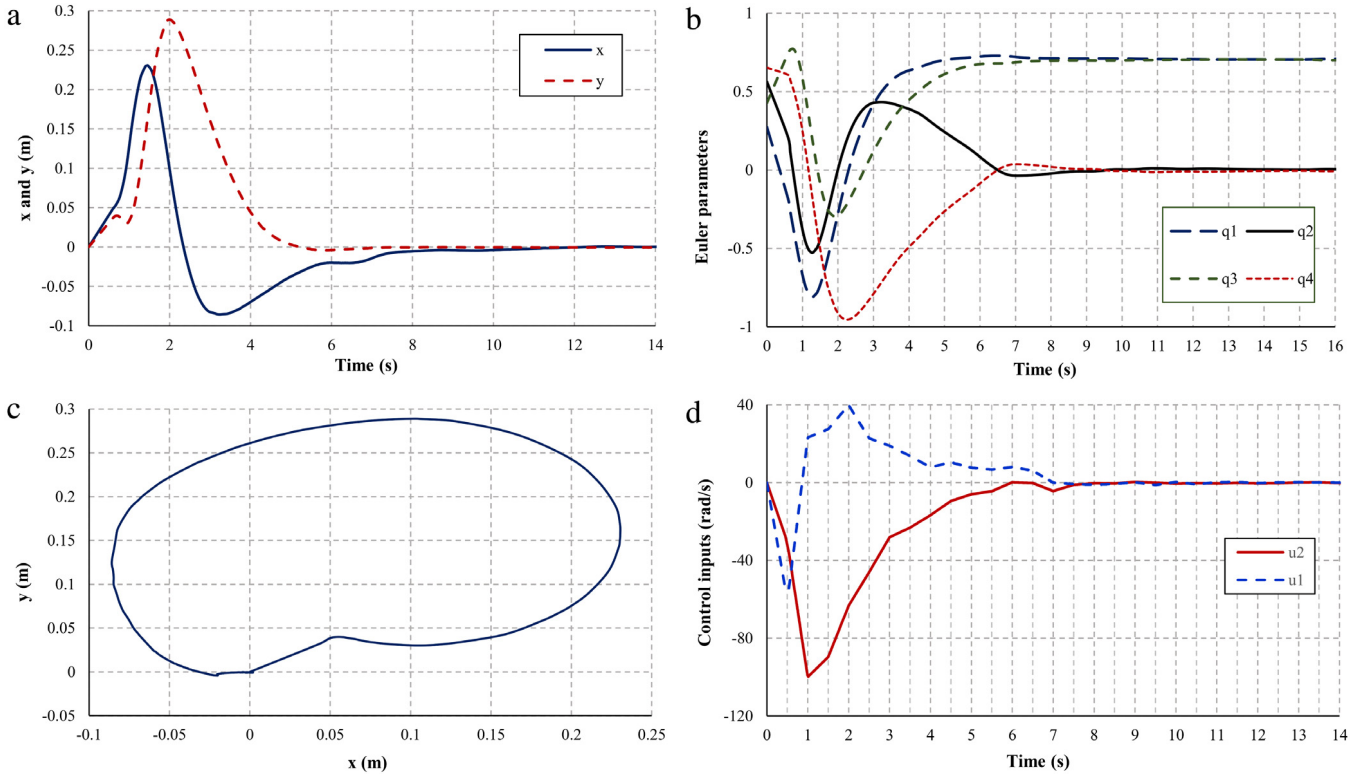


**Fig. 6.** Simulation results for case 4, a: position of the spherical shell, b: Euler's parameters, c: robot's trajectory on the contact plane and d: actuators' input.

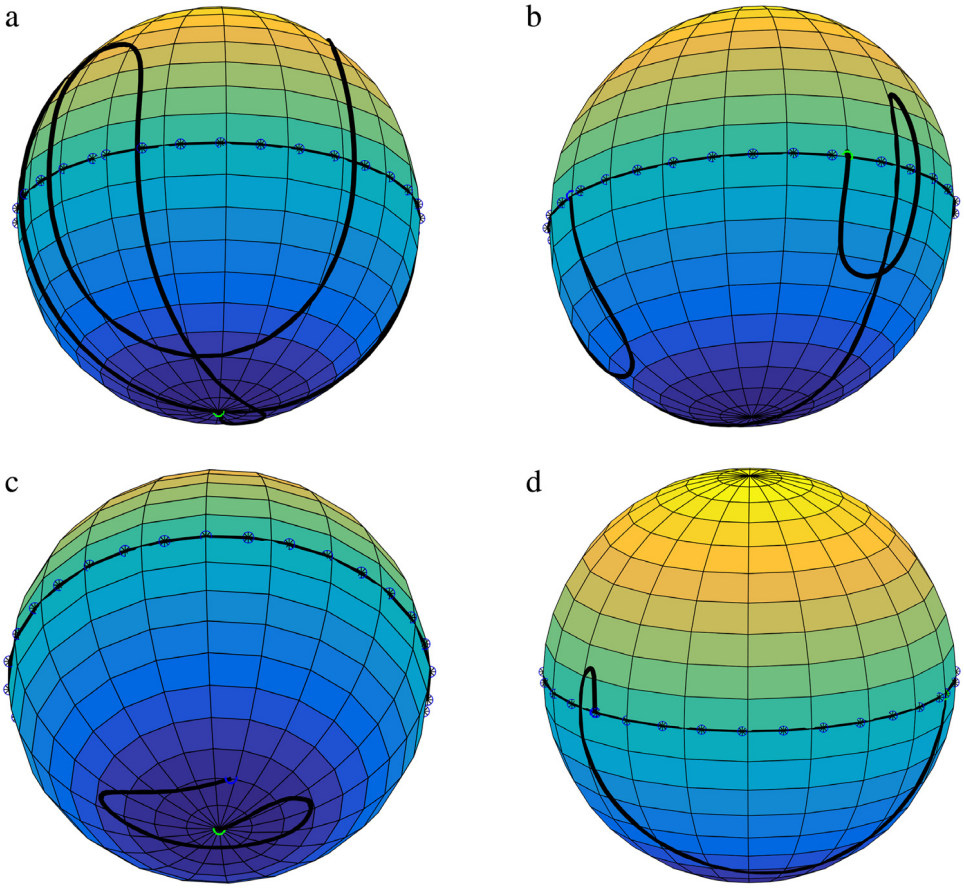


**Fig. 7.** The locus of the contact points of the robot with the ground on the spherical shell, a: case 1, b: case 2, c: case 3 and d: case 4.

interval, ${}^3\dot{\boldsymbol{\omega}}_3$ and $\dot{\mathbf{u}}$ are calculated as follows.

$$\begin{aligned} {}^3\dot{\boldsymbol{\omega}}_3 &\simeq \left({}^3\boldsymbol{\omega}_3\,(k+1) - {}^3\boldsymbol{\omega}_3\,(k)\right)/T; \\ & t_k \le t \le t_{k+1} \\ \dot{\mathbf{u}} = \dot{\mathbf{u}}_k\,(t) &= (\mathbf{u}\,(k+1) - \mathbf{u}\,(k))/T; \\ & t_k \le t \le t_{k+1} \end{aligned} \tag{46}$$

Using (46) the motors' torque are calculated which their values are less than 3.2 N m in the above simulations for $u_{\max} = 100$ rad/s. It is clear that using the smaller value for $u_{\max}$ leads to smaller value for the required torques. However, by decreasing the maximum allowable actuators' input, the required time to stabilize the robot in the desired state may be increased.

- Although the effects of unmodeled dynamics are not considered in the stability analysis of the proposed NMPC system, it is shown that NMPC is inherently robust against small disturbances and uncertainties [53]. The additive disturbance in (43) is considered to compensate unmodeled dynamics including fluctuations of the actuators' input, sudden slipping motions of the spherical shell, small vibrations of the rotors due to misalignment etc. that are neglected to simplify the robot's mathematical model. According to Fig. 6, the disturbances are well compensated by the designed control system which confirms the effectiveness and the robustness of the designed NMPC against the disturbances and uncertainties.
- Although the fmincon MATLAB function has been fast enough in the above simulations for solving the optimal control problem (25) in each sampling time, it is not very suitable for the real time applications. However, several algorithms have been reported in the literature that help to decrease the calculation time of NMPC significantly to use in real time applications [48,54,55]. Furthermore, there exist many embedded systems like Beaglebone black and other ARM based micro-controller boards which are suitable for practical implementation of the NMPC algorithm of the spherical robot. According to our experience, the proposed NMPC for the robot could be executed real-time at many hundreds Hertz [56].

## 5. Conclusion

The stabilization, simultaneous position and orientation control, of a nonholonomic spherical mobile robot actuated by two internal rotors were investigated in this paper. To consider the dynamical realizability conditions of motion, the mathematical model of the robot was derived using angular momentum conservation principle without any simplifying assumption. Through evaluating the controllability of the obtained model, the uncontrollable configurations together with their geometrical meaning were specified. Owing to the nonholonomic structure, the classical control techniques could not be used for perfect stabilization of the robot. Therefore, based on the discretized model of the robot, for the first time a NMPC system has been designed which stabilizes the spherical shell to every desired position and angular orientation. Using the FOH sampling in discretization of the control inputs, a continuous and time-varying feedback control law is obtained from the NMPC which is proposed as a significant technique of stabilizing the nonholonomic systems. The finite horizon stability of the control system was guaranteed through using Lyapunov direct method. The stability analysis demonstrated that by the proper choice of the control system parameters, the stabilization of the closed loop system to a sufficiently small vicinity of every desired position and orientation is guaranteed even in the uncontrollable region. Finally, the performance of the NMPC in different situations was evaluated by computer simulations. The simulation results showed the significant performance of the control system in stabilization of the spherical robot from every initial state to every desired configuration even in the presence of bounded disturbance and parameters' uncertainty.

## References


[1] F. Alouges, Y. Chitour, R. Long, A motion-planning algorithm for the rolling-body problem, IEEE Trans. Robot. 26 (5) (2010) 827–836.
[2] T. Ylikorpi, J. Suomela, Ball-shaped robots, in: H. Zhang (Ed.), Climbing and Walking Robots: Towards New Applications, InTech, 2007, pp. 235–256.
[3] S. Gajbhiye, R.N. Banavar, The Euler-Poincaré equations for a spherical robot actuated by a pendulum, in: IFAC Proceedings, Bertinoro, Italy, 2012.
[4] S. Mahboubi, M.M.S. Fakhrabadib, A. Ghanbaria, Design and implementation of a novel hybrid quadruped spherical mobile robot, Robot. Auton. Syst. 61 (2) (2013) 184–194.
[5] W. Chen, C. Chen, J. Tsai, J. Yang, P. Lin, Design and implementation of a ball-driven omnidirectional spherical robot, Mech. Mach. Theory 68 (2013) 35–48.
[6] F.R. Hogan, J.R. Forbes, Modeling of spherical robots rolling on generic surfaces, Multibody Syst. Dyn. 35 (1) (2015) 91–109.
[7] S. Gajbhiye, R.N. Banavar, Geometric modeling and local controllability of a spherical mobile robot actuated by an internal pendulum, Internat. J. Robust Nonlinear Control 26 (11) (2016) 2436–2454.
[8] R.W. Brockett, Asymptotic stability and feedback stabilization, in: H. Sussmann (Ed.), Differential Geometric Control Theory, 1983, pp. 181–191.
[9] C. Louembet, F. Cazaurang, A. Zolghadri, Motion planning for flat systems using positive B-splines: An LMI approach, Automatica 46 (8) (2010) 1305–1309.
[10] Z. Liang, C. Wamg, Robust stabilization of nonholonomic chained form systems with uncertainties, Acta Automat. Sinica 37 (2) (2011) 129–142.
[11] C.P. Tang, P.T. Miller, V.N. Krovi, J. Ryu, S.K. Agrawal, Differential-flatness-based planning and control of a wheeled mobile manipulator—Theory and experiment, IEEE/ASME Trans. Mechatronics 16 (4) (2011) 768–773.
[12] H. Yuan, Z. Qu, Smooth time-varying pure feedback control for chained non-holonomic systems with exponential convergent rate, IET Control Theory Appl. 4 (7) (2010) 1235–1244.
[13] D. Liu, H. Sun, Q. Jia, A family of spherical mobile robot: Driving ahead motion control by feedback linearization, in: 2nd International Symposium on Systems and Control in Aerospace and Astronautics, Shenzhen, 2008.
[14] M. Kamaldar, M.J. Mahjoob, H. Vahid Alizadeh, Robust speed control of a spherical robot using ARX uncertain modeling, in: IEEE International Symposium on Robotic and Sensors Environments, ROSE, Montreal, 2011.
[15] M. Yue, B. Liu, Adaptive control of an underactuated spherical robot with a dynamic stable equilibrium point using hierarchical sliding mode approach, Internat. J. Adapt. Control Signal Process. 28 (6) (2014) 523–535.
[16] M. Yue, B. Liu, X. Wei, P. Hu, Adaptive sliding-mode control of spherical robot with estimated rolling resistance, Cybern. Syst. Internat. J. 45 (5) (2014) 407–417.
[17] E. Kayacan, E. Kayacan, H. Ramon, W. Saeys, Adaptive neuro-fuzzy control of a spherical rolling robot using sliding-mode-control-theory-based online learning algorithm, IEEE Trans. Cybern. 43 (1) (2013) 170–179.
[18] T. Yu, H.X. Sun, Q.X. Jia, Y.H. Zhang, W. Zhao, Sliding mode control of pendulum-driven spherical robots, in: Z. L, C. Zhang, J. Horng, Z. Chen (Eds.), Advanced Materials Research, 2012, pp. 1519–1522.
[19] Y. Cai, Q. Zhan, X. Xi, Neural network control for the linear motion of a spherical mobile robot, Internat. J. Adv. Robot. Syst. 8 (4) (2011) 79–87.
[20] E. Kayacan, E. Kayacan, H. Ramon, W. Saeys, Velocity control of a spherical rolling robot using a grey-PID type fuzzy controller with an adaptive step size, in: IFAC Proceedings, 2012.
[21] S. Bhattacharya, S.K. Agrawal, Spherical rolling robot: A design and motion planning studies, IEEE Trans. Robot. Autom. 16 (6) (2000) 835–839.
[22] V.A. Joshi, R.N. Banavar, R. Hippalgaonkar, Design and analysis of a spherical mobile robot, Mech. Mach. Theory 45 (2010) 130–136.
[23] V.A. Joshi, R.N. Banavar, Motion analysis of a spherical mobile robot, Robotica 27 (2009) 343–353.
[24] R. Mukherjee, M.A. Minor, J.T. Pukrushpan, Motion planning for a spherical mobile robot: Revisiting the classical ball-plate problem, J. Dyn. Syst. Meas. Control 124 (4) (2002) 502–511.
[25] M. Roozegar, M.J. Mahjoob, M. Jahromi, Dp-based path planning of a spherical mobile robot in an environment with obstacles, J. Franklin Inst. B 351 (2014) 4923–4938.
[26] M. Roozegar, M.J. Mahjoob, M. Jahromi, Optimal motion planning and control of a nonholonomic spherical robot using dynamic programming approach: simulation and experimental results, Mechatronics 39 (2016) 174–184.
[27] Z. Minghui, Z. Qiang, L. Jinkun, C. Yao, Control of a spherical robot: Path following based on nonholonomic kinematics and dynamics, Chin. J. Aeronaut. 24 (2011) 337–345.
[28] Y. Cai, Q. Zhan, C. Yan, Two-state trajectory tracking control of a spherical robot using neurodynamics, Robotica 30 (2012) 195-2.3.

[29] Z. Qiang, L. Zengbo, C. Yao, A back-stepping based trajectory tracking controller for a non-chained nonholonomic spherical robot, Chin. J. Aeronaut. 21 (2008) 472–480.
[30] E. Kayacan, Z.Y. Bayraktaroglu, W. Saeys, Modeling and control of a spherical rolling robot: a decoupled dynamics approach, Robotica 30 (2012) 671–680.
[31] Y. Cai, Q. Zhan, X. Xi, Path tracking control of a spherical mobile robot, Mech. Mach. Theory 51 (2012) 58–73.
[32] T. Otani, T. Urakubo, S. Maekawa, H. Tamaki, Position and attitude control of a spherical rolling robot equipped with a gyro, in: 9th IEEE International Workshop on Advanced Motion Control, Istanbul, 2006.
[33] T. Urakubo, M. Monno, S. Maekawa, H. Tamaki, Dynamic modeling and controller design for a spherical rolling robot equipped with a Gyro, IEEE Trans. Control Syst. Technol. 24 (5) (2016) 1669–1679.
[34] V. Muralidharan, A.D. Mahindrakar, Geometric controllability and stabilization of spherical robot dynamics, IEEE Trans. Automat. Control 60 (10) (2015) 2762–2767.
[35] H. Karimpour, M. Keshmiri, M. Mahzoon, Stabilization of an autonomous rolling sphere navigating in a labyrinth arena: A geometric mechanics perspective, Systems Control Lett. 61 (2012) 495–505.
[36] S. Gajbhiye, R.N. Banavar, Geometric tracking control for a nonholonomic system: a spherical robot, in: IFAC Proceedings, 2016.
[37] A. Morinaga, M. Svinin, M. Yamamoto, A motion planning strategy for a spherical rolling robot driven by two internal rotors, IEEE Trans. Robot. 3 (4) (2014) 993–1002.
[38] M. Svinin, A. Morinaga, M. Yamamoto, On the dynamic model and motion planning for a spherical rolling robot actuated by orthogonal internal rotors, Regul. Chaotic Dyn. 18 (1–2) (2013) 126–143.
[39] K.W. Spring, Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: A review, Mech. Mach. Theory 21 (5) (1986) 365–373.
[40] R.A. Wehage, Quaternions and Euler parameters-a brief exposition, in: E. Haug (Ed.), Computer Aided Analysis and Optimization of Mechanical System Dynamics, Springer, Berlin, 1984, pp. 147–180.
[41] R.M. Murray, Z. Li, S.S. Sastry, A Mathematical Introduction to Robotic Manipulation, CRC, 1994.
[42] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, Constrained model predictive control: Stability and optimality, Automatica 36 (2000) 789–814.
[43] Y. Zhu, U. Ozguner, Constrained model predictive control for nonholonomic vehicle regulation problem, in: IFAC Proceedings, Seoul, 2008.
[44] C. Liu, W. Chen, J. Andrews, Trajectory tracking of small helicopters using explicit nonlinear MPC and DOBC, in: IFAC Proceedings, Milano, 2011.
[45] M. Farrokhsiar, G. Pavlik, H. Najjaran, An integrated robust probing motion planning and control scheme: A tube-based MPC approach, Robot. Auton. Syst. 61 (2013) 1379–1391.
[46] S. Wen, G. Qin, B. Zhang, H.K. Lam, Y. Zhao, The study of model predictive control algorithm based on the force/position control scheme of the 5-DOF redundant actuation parallel robot, Robot. Auton. Syst. 79 (2016) 12–25.
[47] G. Tartaglione, E. D'Amato, M. Ariola, P.S. Rossi, T.A. Johansen, Model predictive control for a multi-body slung-load system, Robot. Auton. Syst. 92 (2017) 1–11.
[48] E. Kayacan, E. Kayacan, H. Ramon, W. Saeys, Learning in centralized nonlinear model predictive control: Application to an autonomous tractor-trailer system, IEEE Trans. Control Syst. Technol. 23 (1) (2015) 197–205.
[49] M. Wang, J. Luo, U. Walter, A non-linear model predictive controller with obstacle avoidance for a space robot, Adv. Space Res. 57 (8) (2016) 1737–1746.
[50] M. Krause, J. Englsberger, P. Wieber, C. Ott, Stabilization of the capture point dynamics for bipedal walking based on model predictive control, in: IFAC Proceedings, 2012.
[51] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, P. Souères, A reactive walking pattern generator based on nonlinear model predictive control, IEEE Robot. Autom. Lett. 2 (1) (2017) 10–17.
[52] L. Grüne, J. Pannek, Nonlinear Model Predictive Control, Springer, London, 2011.
[53] J.B. Rawlings, D.Q. Mayne, Model Predictive Control: Theory and Design, Nob Hill Publishing, Madison, 2009.
[54] M. Vukov, S. Gros, G. Frison, K. Geebelen, J.B. Jorgensen, J. Swevers, Real-time nonlinear MPC and MHE for a large-scale mechatronic application, Control Eng. Pract. 45 (2015) 64–78.
[55] M. Diehl, H.G. Bock, J.P. Schloder, R. Findeisen, Z. Nagy, F. Allgower, Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations, J. Process Control 12 (2002) 577–585.
[56] H. Nourmohammadi, J. Keighobadi, Fuzzy adaptive integration scheme for low-cost SINS/GPS navigation system, Mech. Syst. Signal Process. 99 (15) (2018) 434–449.

**Mahmood Reza Azizi** received the B.Sc. degree in Mechanical Engineering from Shahid Bahonar University. Kerman, Iran, in 2006, and M.S. degrees in Mechanical Engineering from the Department of Mechanical Engineering, Bu Ali Sina University in Hamedan, Iran, in 2008. He is currently a Ph.D. candidate of Mechanical engineering and Control Systems in the faculty of mechanical engineering of Tabriz University, Tabriz, Iran. His research interests include dynamics, motion planning and control of robotic systems. He also has interest in design and implementation of nonlinear control systems, intelligent control and model predictive control.

**Jafar Keighobadi** received the B.Sc. degree in Mechanical Engineering from the University of Tabriz,Tabriz, Iran, in 1997, and MS and Ph.D. degrees in Mechanical Engineering and Control Systems from the Department of Mechanical Engineering, Amirkabir University of Technology (Tehran Polytechnic) in Tehran, Iran, in 2000 and 2008, respectively. He joined the Faculty of Mechanical Engineering, Tabriz University, Iran as an Assistant Professor in 2008. His research interests include estimation and control of stochastic nonlinear systems and the applications in robotic and inertial navigation systems. He also has interest in design and implementation of intelligent control and estimation problems, fault diagnosis and fault-tolerant systems, integrated inertial navigation technology, model predictive control and nonlinear $H_\infty$ filtering/control.