Using Dynamic Programming for Path Planning of a Spherical Mobile Robot

M. Roozegar[1], M. J. Mahjoob[2], and A. Shafiekhani[3]
Center for Mechatronics and Automation, School of Mechanical Engineering
College of engineering, University of Tehran
Tehran, Iran
[1]mroozegar@ut.ac.ir, [2]mmahjoob@ut.ac.ir, [3]shafiekhani@ut.ac.ir

***Abstract*— Spherical mobile robot has good static and dynamic stability, which allows the robot to face different kinds of obstacles and moving surfaces, but the lack of effective control methods has hindered its application and development. In this paper, we propose a direct approach to path planning of a 2-DOFs (Degrees of Freedom) spherical robot based on Bellman's "Dynamic Programming" (DP). While other path planning schemes rely on pre-planned optimal trajectories and/or feedback control techniques, in DP approach there is no need to design a control system because DP yields the optimal control inputs in closed loop or feedback form i.e. after completing DP table, for every state in the admissible region the optimal control inputs are known and the robot can move toward the final position. This enables the robot to function in semi- or even non-observable environments. Results from many simulated experiments show that the proposed approach is capable of adopting an optimal path towards a predefined goal point from any given position/orientation in the admissible region.**

Keywords*— spherical mobile robot; path planning; Dynamic Programming.

I. INTRODUCTION

In recent years, the spherical mobile robot as a member of the new type of mobile robots has made its debut, which consists of a ball-shaped outer shell to include all its mechanism, control devices and energy sources in it. There are many advantages to the use of spherical robot designs, for example the spherical shape allows the robot to face different kinds of obstacles and moving surfaces, since a rolling ball naturally follows the trajectory of least resistance and they cannot be overturned. So they have the advantage to survive in such unmanned or hazardous environment as outer planets, deserts and earthquake ruins, to do some exploration or reconnaissance tasks [1-6]. The status of the design of spherical rolling robots is reviewed in [7].

From the perspective of control, spherical robot is a kind of non-holonomic system that can control more degrees of freedom with fewer drive inputs. Due to the complexity of its control problems, there still have no effective control methodologies for spherical robot, although some researchers devoted significant work. A three-step algorithm to solve the path planning of a sphere rolling on a flat surface was proposed by Li and Canny [8]. Using individual control inputs, two algorithms were presented for partial and complete reconfiguration in [9]. The first strategy uses spherical triangles to bring the sphere to a desired position with a desired orientation. The second strategy uses a specific kinematic model and generates a trajectory comprised of straight lines and circular arc segments. Mukherjee and Das et al. [10] proposed two computationally efficient path planning algorithms for a rolling sphere. Moreover, reconfiguration of a rolling sphere with the perspective of evolute–involute geometry was given in [11]. Shourov Bhattacharya and Sunil K. Agrawal deduced the first-order mathematical model of a kind of spherical robot from the non-slip constraint and the conservation of angular momentum and studied its trajectory planning based on the strategy of optimal time and energy, simulations and experiment results were presented [12, 13]. Bicchi, et al. [14, 15] established a simplified dynamic model for a spherical robot and discussed its path planning on a plane with obstacles. Joshi and Banavar et al. deduced the kinematics model of a spherical robot using Euler parameters and investigated the path planning problems [16, 17]. Based on the Ritz approximation theory, the near-optimal trajectory of BHQ-2 was planned with the Gauss-Newton algorithm in [6].

According to the above review, the motion analysis of spherical robot have been considered in two different aspects: first is the path planning of spherical robot which means finding an optimal trajectory between the initial and final states and second is tracking the desired trajectory via controlling motion of the robot [18, 19].

In this paper, path planning of a 2-DOFs pendulum-driven spherical mobile robot is considered using Bellman's Dynamic Programming (DP). DP is used to find an optimal trajectory between the initial and final positions via finding the corresponding control inputs. In our proposed method, there is no need to design a control system because DP yields the optimal control inputs in closed loop or feedback form i.e. for every state in the admissible region the optimal control inputs are known and the robot can move toward the final position.

The outline of the paper is as follows. The kinematic model of the spherical robot is described in section 2. In section 3, a brief description of DP and its implementation for the spherical robot are presented. Section 4 provides simulation results through MATLAB.

II. KINEMATIC MODEL OF THE SPHERICAL ROBOT

This paper focuses on a simple spherical robot that can be developed using a pendulum based design. Fig. 1 shows a schematic of the internals of a robot called Rotundus. This design consists of two motors. One of them is attached to the horizontal axis that goes through the sphere. In the center there is a pendulum that drops down. When the motor is activated, the sphere will move as long as the weight of the pendulum has enough inertia that it is easier for the casing to spin than the pendulum to go around. The pendulum can move to the left and right by the other motor, causing the robot to turn. So it has a pendulum with 2 DOFs [7].

Consider the motion of a sphere on a flat plate (rolling without slipping). The contact point can be represented as follows in the coordinate system attached to the sphere center.

$$f(\theta,\phi) = \begin{pmatrix} \rho cos\phi cos\theta \\ \rho cos\phi sin\theta \\ \rho sin\phi \end{pmatrix} \tag{1}$$

where ρ,ϕ,θ are variables specifying a point in the spherical coordinates. The contact trajectory on the plate is specified by C=(x,y) in the xyz-coordinates attached to the plane. The contact trajectory on the sphere is indicated by C'= (θ,ϕ) in the coordinates attached to the sphere. In Cartesian coordinates, the rotation angle of the sphere with respect to the surface is the angle between two coordinate systems at the contact point (see Fig. 2). This angle is known as the "holonomy angle", indicated by ψ. The solution of the forward problem in which we seek C with knowledge of C' is obtained as follows [20, 21]:

$$\begin{cases} \dot{x} = \rho\left(-\dot{\theta}\cos\phi\sin\psi + \dot{\phi}\cos\psi\right) \\ \dot{y} = \rho\left(\dot{\theta}\cos\phi\cos\psi + \dot{\phi}\sin\psi\right) \\ \dot{\psi} = -\dot{\theta}sin\,\phi \end{cases} \tag{2}$$

where the input variables are θ and φ. These variables represent a trajectory on the sphere. Since the inputs needed to drive the spherical robot are θ and φ, equation (2) corresponds to the forward kinematics of the robot.

The robot speed is low enough to neglect the dynamics involved, thus the sphere- plate contact point stays always underneath the mass suspended down the pendulum. Therefore, by determining θ and φ, the center of mass and consequently the position of the contact point could be determined from equation (1). As explained before, the motors position the pendulum inside the sphere to change the mass center for a desired move. Thus by changing θ and φ we can find a trajectory which shows the contact point trajectory on the sphere. Then, using equation (2) we can find x and y which represent the contact point trajectory on the plate.

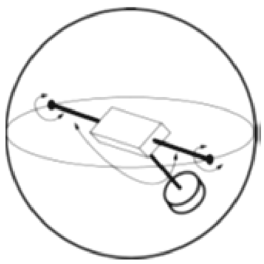

Fig.1. Schematic of the intervals of a pendulum-driven spherical robot

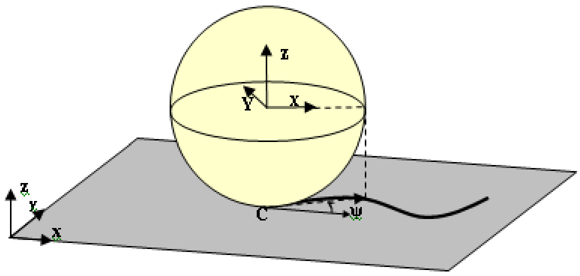

Fig. 2. Holonomy angle in spherical robot

III. Path Planning Using DP

Once the performance measure (cost) for a system has been chosen, the next task is to determine a control function that minimizes this criterion. One of the methods of accomplishing the minimization is Dynamic Programming developed by R. E. Bellman. This method leads to a functional equation that is amenable to solution by digital computer.

The following subsection gives a brief description of DP and implementation of it to our path planning problem is explained after [22, 23, 24].

A. Description of DP

The closed loop or feedback optimal control equation is called the optimal control law, or the optimal policy. The optimal control law specifies how to generate the control value at time *t* from the state value at time *t*. In DP an optimal policy is found by employing the intuitively appealing concept called the principle of optimality. This principle states that "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." DP is a computational technique which extends the decision-making concept to *sequences* of decisions which together define an optimal policy and trajectory. In other words, instead of trying all allowable paths leading from each state to the final state and selecting one with lowest cost which is an exhaustive search, DP considers the application of the principle of optimality. To formulize the computational algorithm for a routing problem the following notation is introduced:

α: *the current state*

u_i: *an allowable decision* (*control*) *elected at the state* α

x_i: *the state adjacent to* α *which is reached by application of* u_i *at* α

h: *the final state*

$J_{\alpha x_i}$: *the cost to move from* α *to* x_i

$J^*_{x_i h}$: *the minimum cost to reach the final state* h *from* x_i

$C^*_{\alpha x_i h}$: *the minimum cost to go from* α *to* h *via* x_i

$J^*_{\alpha h}$: *the minimum cost to go from* α *to* h (*by any allowable path*)

$u^*(\alpha)$: *the optimal decision* (*control*) *at* α

By using this notation the principle of optimality implies that

$$C^*_{\alpha x_i h} = J_{\alpha x_i} + J^*_{x_i h} \quad (3)$$

and the optimal decision at α,u^* (α), is the decision that leads to

$$J^*_{\alpha h} = min\{C^*_{\alpha x_1 h}, C^*_{\alpha x_2 h}, \dots, C^*_{\alpha x_i h}, \dots\}. \quad (4)$$

These two equations define the algorithm Dynamic Programming. The routing problem should be "solved" in a table, where only the consequences of lawful decisions are included. Once the table has been completed, the optimal path from any state to the final state can be obtained by entering the table at the appropriate state and reading off the optimal heading at each successive state along the trajectory. Some characteristics of DP solution are as follows:

- Since a direct search is used to solve the functional recurrence equation, the solution obtained is the absolute (or global) minimum. DP makes the direct search feasible because instead of searching among the set of all admissible controls that cause admissible trajectories, we consider only those controls that satisfy an additional necessary condition-the principle of optimality.

- DP yields the optimal control in closed loop or feedback form i.e. for every state value in the admissible region the optimal control is known.

- DP uses the principle of optimality to reduce dramatically the number of calculations required to determine the optimal control law.

B. Implementation of DP for path planning of spherical robot

In this subsection, the process of implementing DP for the spherical robot to find a set of optimal motions to reach the final position is explained. For convenience dimensionless kinematic equations are obtained and used here.

Assuming $\hat{t} = \frac{t}{T}$, $\hat{x} = \frac{x}{\rho}$ and $\hat{y} = \frac{y}{\rho}$ we have:

$$t = T\hat{t}$$
$$x = \rho\hat{x}$$
$$y = \rho\hat{y}$$
$$\dot{x} = \frac{dx}{dt} = \frac{d(\rho\hat{x})}{d(T\hat{t})} = \frac{\rho}{T}\frac{d\hat{x}}{d\hat{t}} = \frac{\rho}{T}\hat{\dot{x}}$$
$$\dot{y} = \frac{dy}{dt} = \frac{d(\rho\hat{y})}{d(T\hat{t})} = \frac{\rho}{T}\frac{d\hat{y}}{d\hat{t}} = \frac{\rho}{T}\hat{\dot{y}}$$
$$\dot{\psi} = \frac{d\psi}{dt} = \frac{d\psi}{d(T\hat{t})} = \frac{1}{T}\hat{\dot{\psi}}$$
$$\dot{\theta} = \frac{d\theta}{dt} = \frac{d\theta}{d(T\hat{t})} = \frac{1}{T}\hat{\dot{\theta}}$$
$$\dot{\phi} = \frac{d\phi}{dt} = \frac{d\phi}{d(T\hat{t})} = \frac{1}{T}\hat{\dot{\phi}} \quad (5)$$

where T is the total duration of motion and ρ is the radius of the spherical shell.

By substituting these equations in the kinematic model of the spherical robot obtained in section II we have:

$$\dot{x} = \rho(-\dot{\theta}\cos\phi\sin\psi + \dot{\phi}\cos\psi)$$
$$\rightarrow \frac{\rho}{T}\hat{\dot{x}} = \rho\left(-\frac{1}{T}\hat{\dot{\theta}}\cos\phi\sin\psi + \frac{1}{T}\hat{\dot{\phi}}\cos\psi\right)$$
$$\xrightarrow{yields} \hat{\dot{x}} = -\hat{\dot{\theta}}\cos\phi\sin\psi + \hat{\dot{\phi}}\cos\psi$$
$$\dot{y} = \rho(\dot{\theta}\cos\phi\cos\psi + \dot{\phi}\sin\psi)$$
$$\rightarrow \frac{\rho}{T}\hat{\dot{y}} = \rho\left(\frac{1}{T}\hat{\dot{\theta}}\cos\phi\cos\psi + \frac{1}{T}\hat{\dot{\phi}}\sin\psi\right)$$
$$\xrightarrow{yields} \hat{\dot{y}} = \hat{\dot{\theta}}\cos\phi\cos\psi + \hat{\dot{\phi}}\sin\psi$$
$$\dot{\psi} = -\dot{\theta}\sin\phi \rightarrow \frac{1}{T}\hat{\dot{\psi}} = -\frac{1}{T}\hat{\dot{\theta}}\sin\phi$$
$$\xrightarrow{yields} \hat{\dot{\psi}} = -\hat{\dot{\theta}}\sin\phi$$
$$\hat{\dot{\psi}} = T\dot{\psi}$$
$$\hat{\dot{\theta}} = T\dot{\theta}$$
$$\hat{\dot{\phi}} = T\dot{\phi} \quad (6)$$

Consider the following equation of the spherical robot

$$\hat{\dot{x}} = -\hat{\dot{\theta}}\cos\phi\sin\psi + \hat{\dot{\phi}}\cos\psi \quad (7)$$

where $\hat{\dot{\theta}}$ and $\hat{\dot{\phi}}$ are the control variables. Before the numerical procedure of DP can be applied, the system differential equation must be approximated by a difference equation. This can be done most conveniently by dividing the time interval $0 \le t \le T$ $(0 \le \hat{t} \le 1)$ into N equal increments, $\Delta t\ (\Delta\hat{t})$. Then we have:

$$\hat{\dot{x}} = \frac{\hat{x}(\hat{t}+\Delta\hat{t}) - \hat{x}(\hat{t})}{\Delta\hat{t}} =$$
$$-\hat{\dot{\theta}}(\hat{t})\cos\phi(\hat{t})\sin\psi(\hat{t}) + \hat{\dot{\phi}}(\hat{t})\cos\psi(\hat{t})$$
$$\xrightarrow{yields} \hat{x}(\hat{t}+\Delta\hat{t}) =$$
$$\hat{x}(\hat{t}) + \Delta\hat{t}\left(-\hat{\dot{\theta}}(\hat{t})\cos\phi(\hat{t})\sin\psi(\hat{t}) + \hat{\dot{\phi}}(\hat{t})\cos\psi(\hat{t})\right) \quad (8)$$

It will be assumed that $\Delta\hat{t}$ is small enough so that control and state variables can be approximated by a piecewise-constant function that changes only at instants $\hat{t} = 0, \Delta\hat{t}, 2\Delta\hat{t}, \dots, (N-1)\Delta\hat{t}$; thus, for $\hat{t} = k\Delta\hat{t}$ $(k = 0, 1, \dots, N-1)$,

$$\hat{x}([k+1]\Delta\hat{t}) =$$
$$\hat{x}(k\Delta\hat{t}) + \Delta\hat{t}\Big(-\hat{\dot{\theta}}(k\Delta\hat{t})\cos\phi(k\Delta\hat{t})\sin\psi(k\Delta\hat{t})$$
$$+ \hat{\dot{\phi}}(k\Delta\hat{t})\Big) \quad (9)$$

$\hat{x}(k\Delta\hat{t})$ is referred to as the kth value of x and is denoted by $\hat{x}(k)$. So the system difference equation can be written

$$\hat{x}(k+1) =$$
$$\hat{x}(k) + \Delta\hat{t}\left(-\hat{\dot{\theta}}(k)\cos\phi(k)\sin\psi(k) + \hat{\dot{\phi}}(k)\cos\psi(k)\right) \quad (10)$$

In a similar manner for the other states we have:

$$\hat{y}(k+1) =$$
$$\hat{y}(k) + \Delta\hat{t}\left(\hat{\dot{\theta}}(k)\cos\phi(k)\cos\psi(k) + \hat{\dot{\phi}}(k)\sin\psi(k)\right)$$
$$\psi(k+1) = \psi(k) + \Delta\hat{t}\left(-\hat{\dot{\theta}}(k)\sin\phi(k)\right)$$
$$\phi(k+1) = \phi(k) + \Delta\hat{t}\left(\hat{\dot{\phi}}(k)\right) \quad (11)$$

The performance measure (cost) to be minimized here is the energy consumed by the stepper motors, because the energy source of the robot is inside the spherical shell. Assuming ω_1 and ω_2 are the angular velocities of stepper motors for shaft and pendulum, respectively. Then the cost function can be defined as follows:

$$J = \int_0^T (\omega_1^2 + \omega_2^2)dt \quad (12)$$

We have the following relations for our spherical robot gears:

$$\frac{\omega_1}{\dot{\theta}} = 8 \rightarrow \omega_1 = 8\dot{\theta}$$

$$\frac{\omega_2}{\dot{\phi}} = 16 \rightarrow \omega_2 = 16\dot{\phi} \qquad (13)$$

So the cost function can be written as:

$$J = \int_0^T (\omega_1^2 + \omega_2^2)dt =$$

$$\int_0^T ((8\dot{\theta})^2 + (16\dot{\phi})^2)dt = 8^2 \int_0^T (\dot{\theta}^2 + 4\dot{\phi}^2)dt \qquad (14)$$

Or we can define it as:

$$J = \int_0^T (\dot{\theta}^2 + 4\dot{\phi}^2)dt \qquad (15)$$

By substituting the dimensionless relations we have:

$$J = \int_0^T (\dot{\theta}^2 + 4\dot{\phi}^2)dt =$$

$$\int_0^1 \left((\frac{1}{T}\dot{\hat{\theta}})^2 + 4(\frac{1}{T}\dot{\hat{\theta}})^2\right)Td\hat{t} = \frac{1}{T}\int_0^1 \left(\dot{\hat{\theta}}^2 + 4\dot{\hat{\phi}}^2\right)d\hat{t} = \frac{1}{T}\hat{J} \quad (16)$$

where $\hat{J}$ is

$$\hat{J} = \int_0^1 \left(\dot{\hat{\theta}}^2 + 4\dot{\hat{\phi}}^2\right)d\hat{t} \qquad (17)$$

and is the dimensionless cost function. In a similar way this function becomes:

$$\hat{J} = \int_0^1 \left(\dot{\hat{\theta}}^2 + 4\dot{\hat{\phi}}^2\right)d\hat{t} =$$

$$\int_0^{\Delta\hat{t}} \left(\dot{\hat{\theta}}(0)^2 + 4\dot{\hat{\phi}}(0)^2\right)d\hat{t} + \int_{\Delta\hat{t}}^{2\Delta\hat{t}} \left(\dot{\hat{\theta}}(\Delta\hat{t})^2 + 4\dot{\hat{\phi}}(\Delta\hat{t})^2\right)d\hat{t} + \cdots$$

$$+ \int_{(N-1)\Delta\hat{t}}^{N\Delta\hat{t}} \left(\dot{\hat{\theta}}([N-1]\Delta\hat{t})^2 + 4\dot{\hat{\phi}}([N-1]\Delta\hat{t})^2\right)d\hat{t} \qquad (18)$$

Or,

$$\hat{J} = \Delta\hat{t}\left[\dot{\hat{\theta}}(0)^2 + 4\dot{\hat{\phi}}(0)^2 + \dot{\hat{\theta}}(1)^2 + 4\dot{\hat{\phi}}(1)^2 + \cdots + \dot{\hat{\theta}}(N-1)^2 + 4\dot{\hat{\phi}}(N-1)^2\right]$$

$$= \Delta\hat{t}\sum_{k=0}^{N-1}\left(\dot{\hat{\theta}}(k)^2 + 4\dot{\hat{\phi}}(k)^2\right) \qquad (19)$$

Now the method of DP can be applied to the path planning problem.

IV. Simulation Results

In this section, some simulations through MATLAB, including the optimal trajectories found by DP, are presented.

Assuming the spherical robot rolls without slipping on a plane and the initial and final states of the robot to be $P_0 = (x_0, y_0, \psi_0) = (0,0,0)$, and $P_f = (x_f, y_f) = (1,1)$,respectively (there is no constraint on the value of ψ_f). Radius of the robot is assumed to be $\rho = 0.2m$, the specified final time is assumed to be $T = 10sec$, $N = 10$ (a ten-stage process) and the admissible values of the state and control variables are constrained as follows:

$$0 \le x(t) \le 1$$
$$0 \le y(t) \le 1$$
$$-\frac{\pi}{2} \le \psi(t) \le \frac{\pi}{2}$$
$$-\frac{\pi}{6} \le \phi(t) \le \frac{\pi}{6}$$
$$0 \le \dot{\theta}(t) \le \frac{\pi}{2}$$
$$-\frac{\pi}{4} \le \dot{\phi}(t) \le \frac{\pi}{4}$$

To limit the required number of calculations, and thereby make computational procedure feasible, the allowable state and control values must be quantized. Here, it will be assumed that the quantized values are as follows:

$$x(t) = 0, 0.1, 0.2, \ldots, 1$$
$$y(t) = 0, 0.1, 0.2, \ldots, 1$$
$$\psi(t) = -1.57, -1.27, \ldots, 1.43$$
$$\phi(t) = -0.52, -0.32, \ldots, 0.47$$
$$\dot{\theta}(t) = 0, 0.1, 0.2, \ldots, 1.5$$
$$\dot{\phi}(t) = -0.7, -0.6, \ldots, 0.7$$

The simulated optimal trajectories through the DP process are as follows. The approximated trajectory is obtained by neglecting the computational "grid" point error. The exact trajectory is obtained by giving the sequence of optimal controls to the system.

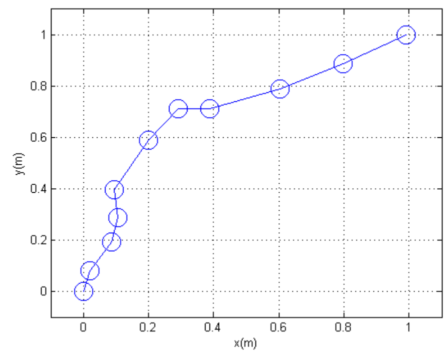

Fig. 1. Approximated trajectory obtained by DP
$P_f = (x_f, y_f) = (0.9919, 1.0012)$

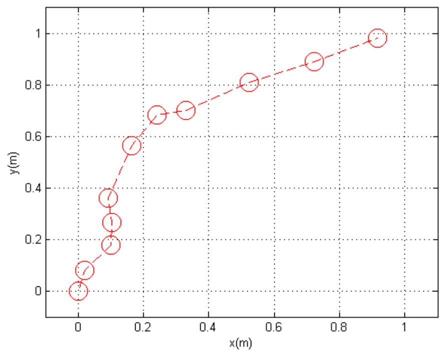

Fig. 2. Exact trajectory obtained by DP
$P_f = (x_f, y_f) = (0.9158, 0.9220)$

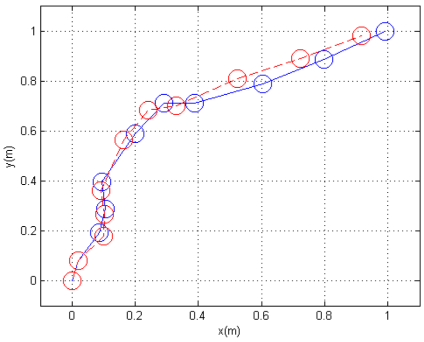

Fig. 3. Comparison between exact and approximated trajectories

It is obvious that the straight line is not the optimal path between the two points because the initial holonomy angle of the robot is important and the cost function should be minimized.

An important point here is that once the DP process reached some optimal paths, the obtained table could be used in any environment even the robot hasn't experienced them. It means that giving any initial and final positions, the robot can follow an optimal path to reach the final position according to DP table.

V. Conclusion

Path planning of a 2-DOFs pendulum-driven spherical mobile robot is considered using Bellman's Dynamic Programming (DP). The robot finds a trajectory with minimum energy consumption through DP. The proposed approach path planning of the robot through DP has a big advantage: there is no need to design a control system because DP yields optimal control inputs in closed loop or feedback form i.e. after completing DP table, for every state in the admissible region the optimal control inputs are known and the robot can move toward the final position. For simulation, the forward kinematic model of robot is used for finding its new position and pose after each motion. Results show that the proposed approach is capable of adopting an optimal path towards a predefined goal point from any given position/orientation in the admissible region.

References

[1] F. cois Michaud and S. Caron, "An Autonomous Toy-Rolling Robot." Proceedings of the 2000 PERCARN-IRIS International Symposium on Robotics (ISR), Montreal Quebec, 2000, pp. 114–119.

[2] R. H. Armour and J. F. V. Vincent, "Rolling in nature and robotics: A review," Journal of Bionic Engineering, vol. 3, pp. 195-208, 2006.

[3] S. Mahboubi, M. M. Seyyed Fakhrabadi, and A. Ghanbari, "Design and Implementation of a Novel Spherical Mobile Robot," Journal of Intelligent & Robotic Systems, pp. 1-22, 2012.

[4] Q. Zhan, T. Zhou, M. Chen, and S. Cai, "Dynamic trajectory planning of a spherical mobile robot," in Robotics, Automation and Mechatronics, 2006 IEEE Conference on, 2006, pp. 1-6.

[5] Y. Cai, Q. Zhan, and X. Xi, "Path tracking control of a spherical mobile robot," Mechanism and Machine Theory, vol. 51, pp. 58-73, 2012.

[6] Q. Zhan, Y. Cai, and Z. Liu, "Near-optimal trajectory planning of a spherical mobile robot for environment exploration," in Robotics, Automation and Mechatronics, 2008 IEEE Conference on, 2008, pp. 84-89.

[7] V. A. Crossley, "A literature review on the design of spherical rolling robots," Pittsburgh, PA, 2006.

[8] Z. Li and J. Canny, "Motion of two rigid bodies with rolling constraint," Robotics and Automation, IEEE Transactions on, vol. 6, pp. 62-72, 1990.

[9] R. Mukherjee, M. A. Minor, and J. T. Pukrushpan, "Simple path planning strategies for spherobot: a spherical mobile robot," in Decision and Control, 1999. Proceedings of the 38th IEEE Conference on, 1999, pp. 2132-2137.

[10] M. A. Minor and J. T. Pukrushpan, "Path planning for a spherical mobile robot: Revisiting the classical ball-plate problem," Ann Arbor, vol. 1001, p. 48109, 2002.

[11] R. Mukherjee, "Reconfiguration of a Rolling Sphere: A Problem in Evolute-Involute Geometry."

[12] S. Bhattacharya and S. K. Agrawal, "Design, experiments and path planning of a spherical rolling robot," in Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on, 2000, pp. 1207-1212.

[13] S. Bhattacharya and S. K. Agrawal, "Spherical rolling robot: A design and path planning studies," Robotics and Automation, IEEE Transactions on, vol. 16, pp. 835-839, 2000.

[14] A. Marigo and A. Bicchi, "A local-local planning algorithm for rolling objects," in Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on, 2002, pp. 1759-1764.

[15] A. Bicchi, A. Balluchi, D. Prattichizzo, and A. Gorelli, "Introducing the "SPHERICLE": an experimental testbed for research and teaching in nonholonomy," in Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on, 1997, pp. 2620-2625.

[16] V. A. Joshi and R. N. Banavar, "Motion analysis of a spherical mobile robot," Robotica, vol. 27, pp. 343-353, 2009.

[17] V. A. Joshi, R. N. Banavar, and R. Hippalgaonkar, "Design and analysis of a spherical mobile robot," Mechanism and Machine Theory, vol. 45, pp. 130-136, 2010.

[18] M. Zheng, Q. Zhan, J. Liu, and Y. Cai, "Control of a Spherical Robot: Path Following Based on Nonholonomic Kinematics and Dynamics," Chinese Journal of Aeronautics, vol. 24, pp. 337-345, 2011.

[19] Z. Qiang, L. Zengbo, and C. Yao, "A back-stepping based trajectory tracking controller for a non-chained nonholonomic spherical robot," Chinese Journal of Aeronautics, vol. 21, pp. 472-480, 2008.

[20] H. V. Alizadeh and M. J. Mahjoob, "Effect of incremental driving motion on a vision-based path planning of a spherical robot," in Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference on, 2009, pp. 299-303.

[21] M. Kamaldar, M. Mahjoob, M. Haeri Yazdi, H. Vahid-Alizadeh, and S. Ahmadizadeh, "A control synthesis for reducing lateral oscillations of a spherical robot," in Mechatronics (ICM), 2011 IEEE International Conference on, 2011, pp. 546-551.

[22] R. E. Bellman, "Dynamic Programming," Princeton, N. J. :Princeton University Press, 1957.

[23] R. E. Bellman, and S. E. Drefus, "Applied Dynamic Programming," Princeton, N. J. :Princeton University Press, 1962.

[24] R. E. Bellman, and R. E. Kalaba, "Dynamic Programming and Modern Control Theory," New York: Academic Press, 1965.