ABSTRACT

Several designs for a spherical rolling robot have been suggested and some of them were implemented. The kinematics and dynamics study as well as the path planning for the rolling robot are based on the assumption that the deterministic model describes the actual rolling robot successfully. However, due to the high mobility of the sphere, the stochastic behavior is obviously observed. In this paper, we first build a rolling robot to confirm the stochasticity. The robot is actuated by a mass-shifting mechanism where an unbalanced weight inside the spherical robot is rotated by two motors, and the imbalance induced by the weight makes the robot roll. After confirming that this actual rolling robot shows the stochastic behavior, we propose a path planning method for the spherical robot rolling on the plane. The path-of-probability method is applied to generate the most probable path from starting location to destination. This planning method uses the stochasticity of the system to produce the probability density function, and generates the piece-wise short steps for the robot move, which construct the whole trajectory that the robot should follow.

1 INTRODUCTION

A spherical rolling robot has drawn considerable attention for its high mobility, simple shape, and interesting application. Example applications of the rolling robot include indoor or outdoor navigation [1], security tasks [2] and human development [3]. Since this robot can be encapsulated completely, it can operate even in the hazardous or dirty environment. While the spherical shape of the robot is easy to roll, accurate actuation and control can become a hard problem, because it can slip as well as roll.

Several actuation schemes for the rolling robot were introduced. The most popular method is to use internal rotors [4, 5] or an unbalanced weight (pendulum) [6, 7]. There exist several unique actuation ideas: a spherical hopping robot [8], a spherical robot with inflatable bladders [9], and linear sliders [10]. In addition, a robust ball wheel mechanism driven by external rollers showed a benefit of the spherical actuator such as omnidirectional motion [11]. The dynamics of the rolling robot was extensively studied [4, 12–15], and a number of references deal with the control problem of the rolling robot [16–19]. In this paper, a method to generate a path and corresponding motion for a rolling robot is proposed. This topic was also covered in many studies [20–23]. However, most studies are performed assuming that the deterministic model perfectly describes the actual system. In theory, the feedback control can compensate the associated error to some extent. However, for example, making a spherical robot completely symmetric in geometry for any orientation is almost impossible. In the stage of path generation, stochastic modeling technique can relieve the problem associated with model error, disturbance, and uncertainty. In this paper, we solve the path (and motion) planning problem for the rolling robot using stochastic modeling and probability approach. To this end, we build a rolling robot, and perform a simple test to observe the stochastic behavior of the robot. Then we apply the path-of-probability (POP) method to generate the path that the robot should follow. This paper is organized as follows. In Section 2, the design and the rolling tests of our rolling robot are presented. In Section 3, the path generation method is explained...
and is adapted to the spherical rolling robot. The stochastic modeling is used in this stage. In Section 4, the examples of paths generated by the proposed method are demonstrated. Finally, the conclusion is provided in Section 5.

2 Robot design and initial tests

2.1 Robot design

We built an initial design of the spherical robot which allows us to observe the stochastic rolling behavior. Our spherical robot (a.k.a. rollbot) shown in Figure 1 is actuated by a pendulum serially connected with a gimbal-like frame which can rotate around two perpendicular axes. This rolling strategy comes from the idea that when the pendulum changes its position, the sphere can roll due to the weight imbalance. Therefore, the spherical robot can move to any direction if the center of the pendulum mass can move toward the corresponding point on the surface of the sphere.

The rollbot with two rotational frames in a hollow sphere is shown in Figure 2. Two axes whose rotation angles are denoted by $\theta$ and $\phi$ can be rotated in either forward or backward directions with geared motors which are expressed as $M_1$ and $M_2$ in the figure. A motor ($M_1$) to rotate the frame by $\theta$ is directly connected to an Xbee receiver attached to an outer rotational frame. A motor ($M_2$) to rotate the pendulum by $\phi$ angle can receive instructions from the receiver attached to the outer rotational frame via photo detectors attached to the inner frame, which can detect infrared light from the IrLEDs attached to the outer rotational frame. Two groups of IrLEDs (one group is for forward rotation and the other group for backward rotation) are attached to the outer frame in the opposite side of the motor ($M_1$) in a concentric circle as shown in Figure 3(b). One group of the circular IrLEDs emits infrared light when the Xbee receiver receives the operation signal for forward rotation while the other group turns on when the operation signal for backward rotation is detected. These groups of IrLED emitters and receivers are used so that the rotational axis of the inner frame can be isolated from the outer frame. Both rotational axes can rotate at any angles without any interference that may be caused by wired connection.

The motors carry 298:1 gear box in front of their shaft. The volume of the motor with gear box is 0.94” x 0.39” x 0.47” without an external shaft of the gear box. The shaft can be rotated at 100 RPM with 70 oz.-inch torques (stall torque) at 6 volt. In terms of appearance, the radius of the spherical shell is 203.2 mm (6 in) and the total weight is 433g. The pendulum mass is 113g, which is experimentally verified to be heavy enough to roll the whole sphere with the offset $d = 15$ mm. The pendulum can roll to any direction except when two rotational axes are put on the singularity position. For example when the pendulum is aligned with the axis of the outer frame (the axis with the angle $\theta$ in Figure 2) by changing the angle $\phi$, it cannot roll to some directions driven by the angle $\theta$ which means a motion of the rollbot caused by the angle $\theta$ cannot influence to the rollbot’s rolling in this case. The communication of the Xbee between control software in PC and the rollbot is performed by IEEE 802.15.4 standard protocol (ZigBee) for which we use a pair of Xbee-s1 modules (Digi International Inc., Minnetonka, MN, USA). One Xbee module is attached to the outer rotational frame of the rollbot, while the other one is connected to the PC with control software. Because this communication is performed by Direct I/O method, there is not any controller equipped to the rollbot. This makes the rollbot small.

2.2 Tests of rolling motion

We tested two different rolling motions: straight rolling (Set #1) and turning direction while rolling on a straight path (Set #2). For the first test (Set #1), we give a constant angular speed $\dot{\theta}$ for the frame with a fixed $\phi$ angle. Ten results are obtained as shown in Figure 4. Figure 4(a) is a result image captured by a camera mounted on the ceiling. The red spots denote
the trace of the rollbot in the ten trials. Figure 4(b) shows the test results from Set #1 as continuous trajectories.

For the second test (Set #2) we added the temporary change on the angle \( \phi \) during the straight rolling. Specifically we rotate the pendulum by \( \phi = 90^\circ \) and then put it back immediately during the rollbot moves on a straight line. Ten results are obtained and shown in Figure 5. In this case, the rollbot changed direction to the left side in the result image. Results for the moving trace and continuous motion plot are shown in Figures 5(a) and 5(b), respectively. Even though the rollbot is forced to roll in a straight line in the Set #1, the rolling traces are laterally perturbed and the rolling directions are not formed vertical lines. Following the ideal motion is impossible practically because of uneven floor, unbalanced robot, and uncertainty in the system and the environment. This opens a way to treat this system as a stochastic one. Although it is true that the stochasticity may ruin the performance of the system, this enables us to apply the probability-based path generation algorithm to this system.

3 Motion and path planning

3.1 Deterministic kinematic model for rolling

Figure 6 shows the rolling sphere and the reference frames for the kinematics of the rolling. Frame \( \{1\} \) is the space frame fixed on the ground. The origin of Frame \( \{2\} \) is placed on the center of the rolling sphere. The z-axis of Frame \( \{2\} \) is always vertical. The rotation angle of Frame \( \{2\} \) is denoted by \( \theta \). Let \( \Omega = [\Omega_x, \Omega_y, \Omega_z]^T \) denote the angular velocity of the rolling measured by Frame \( \{1\} \). The angle \( \theta \) is determined as \( \frac{d\theta}{dt} = \Omega_z \). The angular velocity of the sphere measured by Frame \( \{2\} \) is given as

\[
\vec{\omega} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix}
\]
Since we ignore slipping in the deterministic model, the velocity of the center of the sphere is computed as 
\[
\vec{v}_{O_2} = \vec{\Omega} \times PO_2 = \left( \begin{array}{c} r \Omega_y \\ -r \Omega_x \\ 0 \end{array} \right)
\]
where \( P \) is the contact point between the sphere and the ground, \( r \) is the radius of the sphere and the operation \( \times \) denotes the cross product. Because the vertical projection of \( O_2 \) on the ground is always \( P \) and the distance between the two points is constant, their velocities are the same. Therefore, for desired path for rolling \( \vec{P} = [P_x, P_y, 0]^T \), we can compute the input angular velocity \( \vec{\Omega} \) for the sphere from \( \vec{v}_{O_2} = \frac{d}{dt} \vec{P} \). In other words, we have
\[
\Omega_x = -\frac{\dot{P}_y}{r} \quad \text{and} \quad \Omega_y = \frac{\dot{P}_x}{r}, \quad (1)
\]
If the system is deterministic and we can control the rolling by \( \Omega_x = 0 \) so that the angle \( \theta \) always stays at zero, then the expressions for the angular velocity are the same as \( \vec{\Omega} = \vec{\omega} \).

It is important to note that Frame \( \{2\} \) is used for convenient path generation and is not a body frame. In other words, the rotation of Frame \( \{2\} \) is not the same as that of the rollbot. After the path is generated, the angular velocity written in the fixed frame is given as (1) and it should be converted to the body frame attached on the rollbot as
\[
\vec{\omega}' = R^T \vec{\omega} \quad (2)
\]
where \( R \) is the rotation matrix of the body frame with respect to the space frame.

### 3.2 Stochastic model

Since actual systems always contain a certain amount of uncertainty or disturbance, it is reasonable to include the stochastic effect in the deterministic model. A stochastic differential equation (SDE) in \( \mathbb{R}^n \) is written as
\[
d\mathbf{x}(t) = \mathbf{h}(\mathbf{x}(t), t)dt + \mathbf{H}(\mathbf{x}(t), t)d\mathbf{W}(t) \quad (3)
\]
where \( \mathbf{x} \in \mathbb{R}^n \) and \( \mathbf{W} \in \mathbb{R}^m \). This can be obtained by perturbing the deterministic system \( d\mathbf{x}/dt = \mathbf{h}(\mathbf{x}, t) \) at every value of time by noise or random forcing denoted by \( \mathbf{W}(t) \) where \( \mathbf{W}(t) \) is a vector of uncorrelated Wiener processes, each with unit strength and \( H \in \mathbb{R}^{n \times m} \) is a matrix that scales and couples these noises.

The system variable \( \mathbf{x}(t) \) can be obtained by integrating the SDE (3), but the result will be different due to the noise term every time the SDE is integrated. Let us consider the probability density function \( \rho(\mathbf{x}, t) \) of \( \mathbf{x}(t) \). The Fokker-Planck equation is a partial differential equation that governs the time evolution of the probability density function. The Fokker-Planck equation corresponding to (3) is written as
\[
\frac{\partial \rho(\mathbf{x}, t)}{\partial t} + \sum_{i=1}^{n} \frac{\partial}{\partial x_i} (h_i(\mathbf{x}, t)\rho(\mathbf{x}, t)) - \frac{1}{2} \sum_{i,j=1}^{n} 1 \frac{\partial^2}{\partial x_i \partial x_j} \left( \sum_{k=1}^{m} H_k H_k^T \rho(\mathbf{x}, t) \right) = 0 \quad (4)
\]
The theory was first founded by Fokker and Planck. The advancement for the methodology can be found in many references [24–27].

An alternative way to obtain the probability density function \( \rho(\mathbf{x}, t) \) is to numerically integrate the SDE (3) a number of times and build the histogram which represents the probability density function. The numerical integration of the SDE can be done by Euler-Maruyama method [28].

As confirmed from the test in Section 2.2, the rolling sphere shows the stochastic behavior. The source of stochasticity includes imperfect modeling, disturbance, and uncertainty in the system and the environment. To include the stochastic factors, we assume that the angular velocity contains the stochastic term as
\[
\vec{\omega} dt = \left( \begin{array}{c} \omega_x dt + \sigma_x dW_x \\ \omega_y dt + \sigma_y dW_y \\\omega_z dt + \sigma_z dW_z \end{array} \right)
\]
where \( dW \) is the increment of unit-strength Wiener process, and \( \sigma \) determines the strength of the Wiener process. Therefore, the infinitesimal changes of the position of the contact point and the spinning angle of Frame \( \{2\} \) are given as
\[
\begin{pmatrix}
    dP_x \\
    dP_y \\
    d\theta
\end{pmatrix} = 
\begin{pmatrix}
    r \sin \theta & r \cos \theta & 0 \\
    -r \cos \theta & r \sin \theta & 0 \\
    0 & 0 & 1
\end{pmatrix} 
\begin{pmatrix}
    \omega_x \\
    \omega_y \\
    \omega_z
\end{pmatrix} dt 
+ 
\begin{pmatrix}
    \sigma_x r \sin \theta & \sigma_x r \cos \theta & 0 \\
    -\sigma_x r \cos \theta & \sigma_x r \sin \theta & 0 \\
    0 & 0 & \sigma_z
\end{pmatrix} 
\begin{pmatrix}
    dW_x \\
    dW_y \\
    dW_z
\end{pmatrix} \quad (5)
\]
which is a stochastic differential equation. The probability density function of \( \rho(P_x, P_y, \theta; t) \) can be obtained by solving the corresponding Fokker-Planck equation or numerically integrating the SDE. The probability density function plays an important roll in the path planning that will be explained in the next subsection.
3.3 Path-of-probability method for path generation

The path-of-probability (POP) method was introduced in [29]. After the method was applied to the inverse kinematics problem, it was also used for path generation problems in non-holonomic needle insertion [30, 31].

Figure 7 shows the schematic representation of the POP algorithm. Let us assume that the point $P_1$ is given as a starting point. The cross mark denotes the desired target point. Then we can compare the two choices for the next small path. After the system arrives at $P_2$, the probability of the future path can be obtained from the stochastic approach that appeared in the previous subsection. Figure 7(b) shows the better choice $P_2$ than Figure 7(a) because it gives the higher probability that the future path may hit the destination.

This general approach can be adapted to our problem as follows. First of all, let us assume that we have a simple input angular velocity $\dot{\omega}_z = -1$, $\dot{\omega}_y = 0$, and $\dot{\omega}_x = 0$. Then for a small time period $\Delta t$, the sphere rolls with the angular velocity in a deterministic way. This small path is a simple $+y$ direction motion and may or may not be the best choice in terms for the probability as we see in Figure 7. Therefore we should compare several options for the next small path.

We suggest the following options for the input angular velocity.

\[ \dot{\omega}_z = -\alpha \text{ or } 0 \text{ or } \alpha. \]

where $\alpha$ is a small angular velocity around the $z$-axis. Note that the rotation angle $\theta$ for Frame (2) for this path becomes $\theta = \omega_z \Delta t$. For each choice for the next path, we compute the probability that the future path can hit the target. As aforementioned, the probability will be estimated using the probability density function.

To obtain the probability density function, the Euler-Maruyama method is applied multiple times, which is essentially the numerical integration of the SDE. For the general SDE (3), the numerical integration is given as [28, 30]

\[ x_{i+1} = x_i + h(x_i, t_i)\Delta t + H(x_i, t_i)\Delta W_i \]

where $x_i = x(t_i)$, $\Delta t = t_{i+1} - t_i$, and $\Delta W_i = W(t_{i+1}) - W(t_i)$. Because $W(t_i)$ is the Wiener process, $\Delta W_i$ can be sampled from a Gaussian distribution with zero mean and standard deviation $\sqrt{\Delta t}$.

With the results of multiple paths from the numerical integration, we can build a probability density function. The probability density function can be estimated as

\[ p(x, y) = \frac{1}{n} \sum_{k=1}^{n} g_k(x, y) \]  

(6)

where

\[ g_k(x, y) = \frac{1}{2\pi s^2} \exp\left(-\frac{(x-x_k)^2}{2s^2} + \frac{(y-y_k)^2}{2s^2}\right) \]

(7)

and $(x_k, y_k)$ is the final position of the $k$'th path from the numerical integration of the SDE. The parameter $s$ is chosen so that the resulting probability density function becomes sufficiently smooth.

4 Examples of path generation

4.1 Probability density function

With the input angular velocity $\dot{\omega}_z = -1$, $\dot{\omega}_y = 0$, and $\dot{\omega}_x = 0$, the deterministic rolling will give the straight motion from $(0,0)$ to $(0, 100)$ for 100 sec with the unit radius of the sphere. However, the actual motion will show the stochastic behavior, and the simulation can be done by numerically integrating the SDE. The examples of the many different paths are shown in Figure 8(a). The probability density function of the final position...
of the path is obtained using (6) and visualized in Figure 8(b). If the obstacle is added to the plane, the probability density function can also be obtained by excluding the paths that pass through the obstacle. One example is shown in Figure 9. The circle in Figure 9(a) represents the obstacle.

In this numerical example, the values of the system parameters such as $\alpha$ and $\sigma$'s were chosen so that the numerically generated trajectory can approximately reflect the empirical one. The rigorous method to determine the parameters in the stochastic model was developed in [32].

4.2 Path generation

Figure 10 shows the example results. In Figure 10(a), the three target points (38,90), (-60,70), and (10,98) are considered. For each target, the POP method is applied independently and the resulting paths for the three targets are drawn in one plot in Figure 10(a) for efficient presentation. The whole path is divided 10 segments for the POP method. For each step, the optimal $\omega$ that brings the highest probability that the future path hits the target is searched. The empirical parameters are chosen as $\alpha = 0.02$, $\sigma_x = \sigma_y = 0.05$ and $\sigma_z = 0.02$. The parameter $s$ for the probability density function in (7) is set as $s = 3$ for sufficiently smooth probability density function. The path is obtained by the POP method, and the angular velocity of the rolling motion is computed using (1) and (2).

In Figure 10(b), the obstacle is considered. To obtain the probability density function, we generate many path by numerically integrating the SDE. If the path is overlapped with the obstacle, it is excluded in the construction of the probability density function. The path generated from this probability density function intrinsically avoids the obstacle.

5 Conclusion

In this paper, we built a rolling robot with an unbalanced weight actuated by two motors. From the simple test, we observed the stochastic behavior of the robot. Based on the observation, we derived the kinematics equation for the rolling using the stochastic modeling technique. The kinematics equation became the stochastic differential equation. Using Euler-Maruyama method, the stochastic differential equation was numerically solved, and an ensemble of solutions formed a probability density function for the rolling paths. Using the path-of-probability method with the probability density function, we generated the path and corresponding motion plan for the rolling robot. We verified the method in several obstacle-free cases and a case with an obstacle using simulation.

The prototype robot was an open-loop system. We did not
equipping any sensors or controllers because the purpose of this prototype robot is to see possibilities of the proposed design concept and the stochastic behavior. Accurate motion control based on the path and motion plan can be achieved by building a closed-loop system with feedback sensors and balancing controllers. We leave this as future work.

REFERENCES


