Controlling a Spherical Robot in Various Configurations

Mateus A. Araujo, Kunal Patel, Ria Pereira, Jeffrey Wilder

Abstract— **Spherical robots have sparked recent interest with the recent advancement of control theory and hardware technology. The control of such robots is already established, but the idea of controlling a stack of spherical robots seems to be an un-touched research area. While limited in applications, such a problem provides a very interesting control theory research topic. In this paper, we detail our research in the control of barycenter offset spherical robots reduced to 2D (hoops) on different surfaces including the trivial flat surface, inclined surface, spherical surface (fixed bottom hoop), and stacked-hoop configurations.**

I. INTRODUCTION AND LITERATURE REVIEW

Control theory has developed quickly in recent years and is fundamental to most Robotic applications. A highly intriguing and novel idea is the concept of a spherical robot. Due to its spherical nature, a spherical robot is holonomic (it can move in any direction, regardless its orientation), is inherently immune to being overturned, can easily be sealed to the elements, and recover from collisions [1]. Spherical robots are ideal for hazardous environments, being able to function in snow, sand, and mud. They can even be made to float and travel through water [2]. They have been used for surveillance solutions (Rotundus GroundBot [3]), agricultural crop monitoring, and entertainment (Sphero [4] and BB-8 [5]).

There is no dominant mechanical design for spherical robots, resulting in a wide range of robotic characteristics, capabilities, and control designs. There are three major mechanism types used to propel spherical robots: Barycenter Offset (BCO), Shell Transformation (ST) and Conservation of Angular Momentum (COAM) [1].

Shell Transformation is a novel approach, where the actuation relies on deformation of the outer body (shell), but this method is mechanically complex and reliability would be a concern. COAM-based spherical robots derive their motive forces by spinning a massive gyro within the sphere, which can be used to produce a torque on the outer sphere. Although this method can be used to exert significantly higher motive torques (since the maximum torque is not dependent on the torque due to the influence of gravity on an offset mass), and consequently make much faster and better performing robots, this is the least used mechanism due to its high complexity [1].

The BCO method relies on shifting the robot's center of mass (the barycenter) in order to produce a torque around the contact point. As its internal mechanisms move from their equilibrium positions, the mass distribution of the ball will be shifted, causing the ball to roll to a new position of equilibrium. This is the most commonly used principle and consequently, is the method we will base our research on [1] [2].

There are many ways to implement spherical robots based on the BCO principle. Due to its simplicity and easy transition from 2D to 3D, a pendulum based approach was used for this project. The mechanism consists of a fixed shaft through the center of the outer shell of the robot, with a pendulum and massive bob that rotates around the shaft. Rotating the pendulum shifts the center of mass outward from the center. This applies a torque to the shell, which begins to roll. Shifting the pendulum left or right causes the robot to steer.

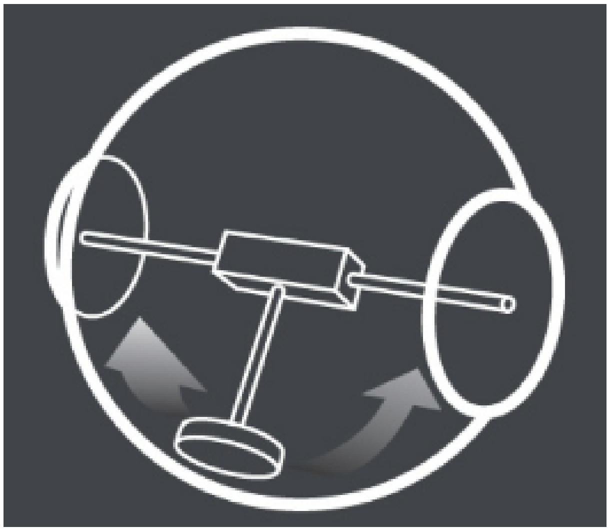


Fig. 1. Pendulum-based BCO-type spherical robot [1]

Current research in the control of spherical robots is well established. Examples of currently existing control techniques can be found in [6] [7] and [8]. However, the current research appears to stop at a single sphere. A highly intriguing extension of the control of a spherical robot would be investigating the control of a stack of spherical robots. The apparent lack of research interest in this area could be attributed to a few different factors, a highly likely scenario being that its limited applications don't warrant much research in the area. However, such a system would be a very interesting show piece, and an even more interesting control problem. Intuition suggests that this problem is highly ambitious for a semester-long class project. Therefore, a more manageable goal was established.

II. Problem Statement

With the end goal of balancing a spherical robot on top of another spherical robot being too ambitious, simplifications must be made. The problem will be reduced from 3D to 2D, considering a spherical robot as a hoop and a pendulum in the middle that can apply torque to the hoop. We will explore the control of this mechanism on multiple surfaces, starting with a flat plane, then a slope of constant angle, and a fixed hoop (spherical surface). Finally, we will investigate controlling a stack of two such hoops.

III. Dynamic Model

For the first three scenarios, the dynamics only have to consider a single hoop, so we can use the same dynamic model. However, when a stack of two controlled hoops is considered, either the coupling of the hoops must be modeled as a disturbance, or an entirely new dynamic model must be produced, considering the stacked-hoop system as a single system. The second approach was chosen.

A. Single Hoop on Slope

To derive the dynamic model of the system, multiple samples of current literature were consulted. [9] and [6] give the most intuitive dynamic models of the system, but each has its limitations. [9] provides a dynamic model of the system considering viscous damping forces for the mechanism and the mass of the drive motor assembly. However, it only considers system working on flat surface. [6] provides dynamics of the system on an inclined surface, but does not consider either the viscous forces or mechanism mass. The model used in our research is derived by combining the dynamics from both references as illustrated in Fig. 2.

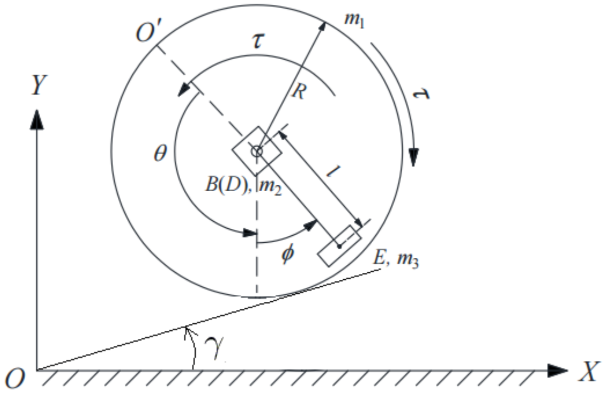


Fig. 2. Hoop With Gamma (Adapted from [9])

Where m_1, m_2, m_3 are the masses of the shell, motor mechanism, and pendulum bob, R, l are the radius of the sphere and the length of the pendulum, θ, ϕ are the sphere's displacement from the origin, and the pendulum's displacement from vertical, I_1, I_2, I_3 are the moments of inertia of the shell, mechanism, and bob, and τ is the torque applied between the pendulum and the sphere.

The dynamic model is derived by calculating the Lagrangian $L = T - P$ where T is kinetic energy of the system and P is the potential energy of the system, as shown below:

$$T = \frac{1}{2}J_1\dot{\theta}^2 + \frac{1}{2}J_2\dot{\phi}^2 + m_3Rl\theta\dot{\phi}cos(\phi)$$

$$P = -m_3glcos(\phi)$$

With the following constants defined:

$$\begin{aligned} M_t &= m_1 + m_2 + m_3 \\ J_1 &= M_tR^2 + I_1 \\ J_2 &= m_3l^2 + I_1 + I_2 \\ g &= GravitationalAcceleration \end{aligned}$$

Note: We believe the last term in the above equation for kinetic energy (T) should have $cos(\phi)$ replaced with $sin(\phi)$ according to [1]. The Langrangian equation, after going through the derivatives, can be expressed in the following form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{E}(\mathbf{q})\tau \tag{1}$$

Where $q = [\theta, \phi]^T$ and:

$$M(q) = \begin{bmatrix} J_1 & m_3Rlcos(\phi)) \\ m_3Rlcos(\phi) & J_2 \end{bmatrix}$$

$$N(q, \ddot{q}) = \begin{bmatrix} \zeta(\dot{\theta} + \dot{\phi}) - m_3Rlsin(\phi)\dot{\phi}^2 \\ \zeta(\dot{\theta} + \dot{\phi}) + m_3glsin(\phi) \end{bmatrix}$$

$$E(q) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Using the state vector $x = [q, \dot{q}]^T = [\theta, \phi, \dot{\theta}, \dot{\phi}]^T$, the dynamic equations can be written as follows:

$$\begin{aligned} \dot{x_1} &= x_3 \\ \dot{x_2} &= x_4 \\ \dot{x_3} &= f_1(x) + b_1(x)u \\ \dot{x_4} &= f_2(x) + b_2(x)u \end{aligned}$$

With the following functions defined:

$$f_1(x) = \frac{m_{22}n_1 - m_{12}n_2}{m_{12}m_{21} - m_{11}m_{22}}$$

$$f_2(x) = \frac{m_{11}n_2 - m_{21}n_1}{m_{12}m_{21} - m_{11}m_{22}}$$

$$b_1(x) = \frac{m_{22} - m_{12}}{m_{11}m_{22} - m_{12}m_{21}}$$

$$b_2(x) = \frac{m_{11} - m_{21}}{m_{11}m_{22} - m_{12}m_{21}}$$

Here, m_{ij} is an element of matrix $M(q)$ and n_k is an element of vector $N(q, \ddot{q})$.

The above model 1 is taken directly from [9] and doesn't account for the slope angle γ. We reference [6] to modify the model slightly. The updated dynamic model replaces ϕ with $\phi - \gamma$ in the kinetic energy equations. However, since the potential energy of the pendulum is measured with respect to gravity and ϕ is measured with respect to gravity, inclination of the surface does not change the potential energy of the pendulum. A new term for the potential energy of the sphere, due to inclination, is added to the equation. The following components of 1 are modifed from above. The rest remains the same.

$$M(q) = \begin{bmatrix} J_1 & m_3 Rlcos(\phi-\gamma)) \\ m_3 Rlcos(\phi-\gamma) & J_2 \end{bmatrix}$$

$$N(q,\ddot{q}) = \begin{bmatrix} \zeta(\dot{\theta}+\dot{\phi}) + M_t gRsin(\gamma) - m_3 Rlsin(\phi-\gamma)\dot{\phi}^2 \\ \zeta(\dot{\theta}+\dot{\phi}) + m_3 glsin(\phi) \end{bmatrix}$$

B. Equilibrium Analysis

In equilibrium analysis, we are only interested in static equilibrium. The system has many static equilibrium points depending on the configuration. For the hoop on a flat surface, equilibrium analysis is trivial, and it is obvious that any point is an equilibrium point, as long as $\phi = 0$. The state vector is as follows: $X^* = [C, 0, 0, 0]^T$. C is any constant value for θ.

For the non-trivial case where the ground (or hoop) below causes the surface to be sloped, the equilibrium point becomes a function of the slope angle. The analysis involves a simple angular momentum balance, and is left to the reader. The results are as follows:

$$\phi_{eq} = arcsin(\frac{M_t Rsin(\gamma)}{m_3 l}) \tag{2}$$

And, the slope as a function of ϕ:

$$\gamma = arcsin(\frac{m_3 lsin(\phi)}{M_t R}) \tag{3}$$

Note that since the maximum torque applied by the pendulum occurs at $\phi = \frac{\pi}{2}$, there is a maximum slope at which the sphere can remain in static equilibrium. This limit is $\gamma = arcsin(\frac{m_3 l}{M_t R})$ which, for our chosen parameters, came out to be $18.76°$.

C. Stacked Hoops

The next challenge was to balance one hoop on top of another hoop and design a control strategy that can translate the bottom hoop while simultaneously balancing the top hoop. The procedure is as follows:

1) Derive the dynamic model of the system using the Lagrangian method
2) Test the uncontrolled dynamic model for intuitive behavior
3) Apply control algorithm with trajectory following

In our literature review, we were unable to find a dynamic model for a stacked-hoop system. Therefore, we derived our own using the Lagrangian method. The dynamic model of the system is derived by first solving the position and velocity kinematics of the centers of mass of the spheres:

As shown in Fig. 3, the state variables for the system are $\theta_1, \theta_2, \phi_1, \phi_1$. The angle δ is calculated from θ_1 and θ_2. The positions of the centers of mass for sphere 2 (bottom) and its pendulum are calculated as follows:

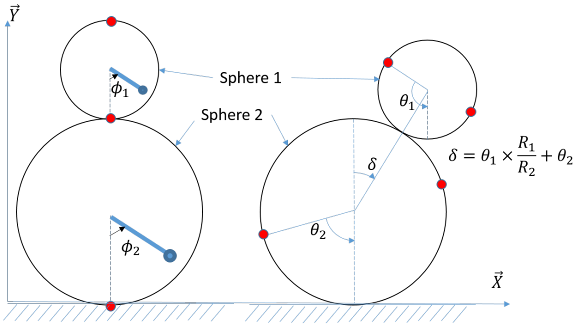


Fig. 3. Notations to derive Dynamics for two hoops

$$\overrightarrow{posSphere}_2 = [R_2\theta_2; R_2; 0]^T$$
$$\overrightarrow{posPend}_2 = \overrightarrow{posSphere}_2 + l_{pend2}[sin(\phi_2); -cos(\phi_2); 0]^T$$

The position of the center of mass of sphere 1 can be found from that of sphere 2:

$$\overrightarrow{posSphere}_1 = \overrightarrow{posSphere}_2 + (R_1+R_2)[sin(\delta_2); cos(\delta_2); 0]^T$$

where $\delta = \theta_1 R_1/R2 + \theta_2$. And the position of pendulum 1:

$$\overrightarrow{posPend}_1 = \overrightarrow{posSphere}_1 + l_{pend1}[sin(\phi_1); -cos(\phi_1); 0]^T$$

These positions vector expressions as functions of θ_1,θ_2,ϕ_1, and ϕ_2. By differentiating with respect to time, the velocities can be found as functions of $\dot{\theta}_1$,$\dot{\theta}_2$,$\dot{\phi}_1$ and $\dot{\phi}_2$.

$$\vec{V}_{s1} = \frac{d}{dt}(\overrightarrow{posSphere}_1); \vec{V}_{p1} = \frac{d}{dt}(\overrightarrow{posPend}_1)$$
$$\vec{V}_{s2} = \frac{d}{dt}(\overrightarrow{posSphere}_2); \vec{V}_{p2} = \frac{d}{dt}(\overrightarrow{posPend}_2)$$

The angular velocities of the spherical shell are given as

$$\omega_1 = (\dot{\theta}_1 + \dot{\theta}_2)[0; 0; 1]^T$$
$$\omega_2 = \dot{\theta}_2[0; 0; 1]^T$$

The kinetic and potential energies (to be used in the Lagrangian) of each mass, can be found as follows:

$$K_1 = \frac{1}{2}m_{s1}\vec{V_{s1}}\cdot\vec{V_{s1}} + \frac{1}{2}I_{s1}\vec{\omega_1}\cdot\vec{\omega_1} + \frac{1}{2}m_{p1}\vec{V_{p1}}\cdot\vec{V_{p1}}$$
$$K_2 = \frac{1}{2}m_{s2}\vec{V_{s2}}\cdot\vec{V_{s2}} + \frac{1}{2}I_{s2}\vec{\omega_2}\cdot\vec{\omega_2} + \frac{1}{2}m_{p2}\vec{V_{p2}}\cdot\vec{V_{p2}}$$

where, I_{s1} and I_{s2} are the moments of inertia of the spheres.

$$P_1 = m_{s1}\vec{g}^T \cdot \overrightarrow{posSphere}_1 + m_{p1}\vec{g}^T \cdot \overrightarrow{posPend}_1$$
$$P_1 = m_{s2}\vec{g}^T \cdot \overrightarrow{posSphere}_2 + m_{p2}\vec{g}^T \cdot \overrightarrow{posPend}_2$$

Then, the Lagrangian of the system is found, and and the dynamic equations are applied to the Lagrangian for each state variable:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\theta_1}}\right) - \frac{\partial L}{\partial \theta_1} &= \tau_1 - \zeta_1(\dot{\phi} + \dot{\theta}) \\
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\phi_1}}\right) - \frac{\partial L}{\partial \phi_1} &= \tau_1 - \zeta_1(\dot{\phi} + \dot{\theta}) \\
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\theta_2}}\right) - \frac{\partial L}{\partial \theta_2} &= \tau_2 - \zeta_2(\dot{\phi} + \dot{\theta}) \\
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\phi_2}}\right) - \frac{\partial L}{\partial \phi_2} &= \tau_2 - \zeta_2(\dot{\phi} + \dot{\theta})
\end{aligned}
$$

These computations are carried out in MATLAB using the symbolic toolbox. The resulting dynamic model contains four equations. MATLAB's equationsToMatrix() function is used to separate the second time derivatives $[\theta_1, \theta_2, \phi_1, \phi_2]$ of the state variables from the Coriolis and gravity terms, yielding the dynamic model in the following form:

$$
M_{4x4}(\phi_1, \phi_2) \begin{bmatrix} \ddot{\theta_1} \\ \ddot{\theta_2} \\ \ddot{\phi_1} \\ \ddot{\phi_2} \end{bmatrix} + C_{4x1}(\theta_1, \theta_2, \phi_1, \phi_2, \dot{\theta_1}, \dot{\theta_2}, \dot{\phi_1}, \dot{\phi_2})
$$

$$
+G_{4x4}(\theta_1, \theta_2) = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix}
$$

D. *Testing the Model*

The model is tested by setting the initial position of the top hoop (or sphere) slightly offset from the apex of the bottom hoop. Then the configuration is allowed to act under gravity without applying any torque to any of the pendulums. These results are then simulated and visually inspected. The damping at the pendulum is supposed to dissipate the kinetic energy and the top hoop is expected to oscillate with a gradually reducing amplitude, eventually settling underneath the top hoop. The simulation results reliably demonstrate this behavior. A graph of the state variables for the uncontrolled free-fall test is shown in Fig. 4:

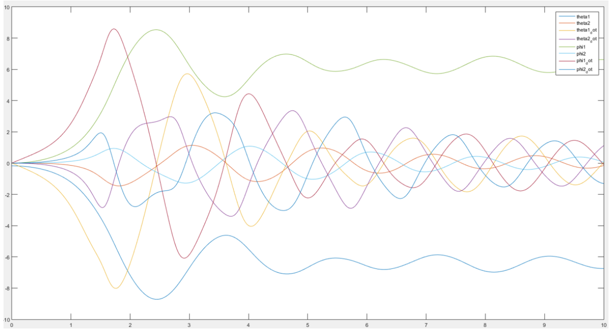


Fig. 4. Plot of state variables under free-fall without any control input. Units: rad and rad/sec on Y axis for angular displacement and velocity state variables. Time in seconds on X axis.

IV. Control

In determining which methods of control to implement, the complexity of the system, current approaches in literature, and topics discussed in class were considered. Classical control was ruled out due to the system complexity. Each control method was first implemented for a system on a flat plane, then on a slope, then on a fixed sphere, and finally, on the stacked-hoop system. The methods explored are described in detail in the following sections:

A. *Partial Feedback Linearization*

Since the system is underactuated, and we are more interested in controlling θ (which is proportional to the linear position: $y = R\theta$), we thought to find a torque that would linearize the error of θ. Therefore, the system should be devided into two sub systems: one for θ, which is controllable, and another for ϕ, whose behavior is unknown. The controller derivation is as follows:

Divide the system into a linearized and non-linearized subsystems:

$$
\begin{aligned}
x_L &= [\theta; \dot{\theta}] \\
\dot{x}_L &= Ax_L \\
x_{NL} &= [\phi; \dot{\phi}] \\
\dot{x}_{NL} &= f(x, x_{NL})
\end{aligned}
$$

Where the eigenvalues of A are negative (stability condition).

The details of this controller have been omitted due the fact it did not produce good results for this application.

B. *Computed Torque Control*

1) Single Hoop (Flat Plane, Inclined Plane, Fixed Hoop): The goal of this controller is to stabilize the error of the system with error feedback and feedforward compensation. To derive this controller we define the error variable:

$$
e = q - q_d \tag{4}
$$

Where q_d is the desired state for $q = [\theta; \phi]$. Through simple manipulations on eq. 1:

$$
M(q)\ddot{e} = E(q)\tau - N(q, \dot{q}) - M(q)\ddot{q}_d \tag{5}
$$

Then we define the desired state for the error:

$$
\begin{gathered}
\ddot{e} = A\dot{e} \\
\begin{bmatrix} \ddot{e}_\theta \\ \ddot{e}_\phi \end{bmatrix} = \begin{bmatrix} -k_{p\theta} & -k_{v\theta} \\ -k_{p\phi} & -k_{v\phi} \end{bmatrix} \begin{bmatrix} \dot{e}_\theta \\ \dot{e}_\phi \end{bmatrix}
\end{gathered} \tag{6}
$$

$k_{p\theta}$, $k_{v\theta}$, $k_{p\phi}$ and $k_{v\phi}$ must be positive to stabilize e. Substituting eq. 6 in eq. 5 we get:

$$
E(q)\tau = M(q)A\dot{e} + N(q, \dot{q}) + M(q)\ddot{q}_d \tag{7}
$$

Since it is not possible to solve τ for both equations (underactuated system), the first value of the τ vector (that is supposed to stabilize θ, which is the main variable to be controlled) is chosen. Then we get:

$$
\begin{aligned}
\boldsymbol{u} =& \boldsymbol{m_{11}(-k_{p\theta}\dot{e}_\theta - k_{v\theta}\dot{e}_\theta) + m_{12}(-k_{p\phi}\dot{e}_\phi - k_{v\phi}\dot{e}_\phi)} \\
&+ \boldsymbol{n_1 + m_{11}\ddot{\theta}_d + m_{12}\ddot{\phi}_d}
\end{aligned} \tag{8}
$$

2) Dual Hoop (Stacked-Hoop Dynamic Model): Applying computed torque control to the stacked-hoop system follows a similar methodology to the single-hoop system. The control Law for this trial was same as that of the computed torque control of the single-hoop system. The torque required for control is computed as follows:

$$\begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\\ \tau_4\end{bmatrix} = M_{4\times 4}\begin{bmatrix}\ddot{e_1}\\ \ddot{e_2}\\ \ddot{e_3}\\ \ddot{e_4}\end{bmatrix} + C_{4x1}(\theta_1,\theta_2,\phi_1,\phi_2,\dot{\theta}_1,\dot{\theta}_2,\dot{\phi}_1,\dot{\phi}_2)$$

$$+G_{4x4}(\theta_1,\theta_2)$$

Where

$$\begin{aligned}\ddot{e_1} &= kp1(\theta_1-\theta_{1d}) + kv1(\dot{\theta}_1-\dot{\theta}_{1d})\\ \ddot{e_2} &= kp2(\theta_2-\theta_{2d}) + kv2(\dot{\theta}_2-\dot{\theta}_{2d})\\ \ddot{e_3} &= kp3(\phi_1-\phi_{1d}) + kv3(\dot{\phi}_1-\dot{\phi}_{1d})\\ \ddot{e_4} &= kp4(\phi_2-\phi_{2d}) + kv4(\dot{\phi}_2-\dot{\phi}_{2d})\end{aligned}$$

After computing the four values of torque, we only pick the torque computed for θ_d for a sphere and use the same torque value for ϕ_d. This is because it is an under actuated system.

C. Robust Control

The robust controller, as described in [15], was adapted to our dynamic system. The dynamic equation as seen in Eq. 1 was re-written to extract the Inertia matrix (M), Coriolis matrix (C) and Gravity matrix (G) as follows:

$$\mathbf{M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q,\dot{q})} = \tau$$

$$M(q) = \begin{bmatrix}J_1 & m_3Rlcos(\phi-\gamma))\\ m_3Rlcos(\phi-\gamma) & J_2\end{bmatrix}$$

$$C(q,\ddot{q}) = \begin{bmatrix}\zeta & \zeta - m_3Rlsin(\phi-\gamma)\dot{\phi}\\ \zeta & \zeta\end{bmatrix}$$

$$G(q,\ddot{q}) = \begin{bmatrix}M_tgRsin(\gamma)\\ m_3glsin(\phi)\end{bmatrix}$$

From this Lagrangian dynamic equation, an inertial parameterization vector, B, was defined as follows:

$$B = \begin{bmatrix}M_tR^2+I_1\\ M_tR\\ m_3Rl\\ m_3l^2+I_3+I_2\\ \zeta\\ m_3l\end{bmatrix}$$

Following the equations of the robust control algorithm as described in [15], the torque is computed, as seen in Eq. 9, such that state variables θ and ϕ follow a cubic polynomial trajectory:

$$\tau = \mathbf{Y(B+u) - Kr} \tag{9}$$

where

$$Y = \begin{bmatrix}Y_1\\ Y_2\end{bmatrix}$$

with

$$Y_1' = \begin{bmatrix}a_1\\ gsin(\gamma)\\ a_2 * cos(\gamma-\phi) + \dot{\phi}v_2sin(\gamma-\phi)\\ 0\\ v_1+v_2\\ 0\end{bmatrix}$$

$$Y_2' = \begin{bmatrix}0\\ 0\\ a_1cos(\gamma-\phi)\\ a_2\\ v_1+v_2\\ gsin(\phi)\end{bmatrix}$$

Next, the gain matrices were set to

$$K = \begin{bmatrix}0.001 & 0\\ 0 & 0.001\end{bmatrix}$$

$$\lambda = \begin{bmatrix}0.001 & 0\\ 0 & 0.001\end{bmatrix}$$

The bound of uncertainty, ρ, was set to 10 and the bound of steady state error at infinity, ϵ, was set to 4.

For these configuration parameters, a single hoop system was simulated first on a horizontal plane and then on an inclined plane.

D. Hierarchical Sliding Mode Control

A prevalent approach in literature on the control of spherical robots is to implement a Hierarchical Sliding Mode Controller[9] [6] [13]. The method described in [12], was followed. Thus, we first introduce Sliding Mode Control and then explain the applied controller in more detail.

1) Introduction to Sliding Mode Control: Sliding Mode Control (SMC) is a powerful and robust nonlinear feedback control method, which is insensitive to system parameter uncertainties and external disturbances. SMC introduces a function known as a "sliding variable". When the sliding variable becomes zero, it defines a sliding surface. Sliding mode controllers try to drive the sliding variable to zero, and use control methods to keep it there[14].

For a non-linear system:

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= f(x) + b(x)u\end{aligned}$$

The following behavior is desired (feedback linearization):

$$\dot{e} = -\lambda e \rightarrow \dot{e} + \lambda e = 0$$

Where $\lambda > 0$ and $e = x - x_d$.

These functions may not be zero all the time, especially with uncertainties, disturbances or in underactuated systems. The sliding variable is defined as follows:

$$s = \dot{e} + \lambda e$$

The structure of the sliding mode controller is as follows according to [12]:

$$u = u_{eq} + u_{sw}$$

Where

$$
\begin{aligned}
u_{eq} &= -\frac{(f(x) + \lambda\dot{e} - \ddot{x}_d)}{b(x)} \\
u_{sw} &= -\eta sgn(S) - kS
\end{aligned}
$$

with $\eta > 0$ and $k > 0$ properly designed.

u_{eq} is called equivalent control, which is designed for when the system states are in the sliding mode $s = 0$ (the derivation is similar to computed torque control for a planar robot arm). u_{sw} is called a switch input, which drives the system towards the sliding mode. In addition, notice that u_{sw} is discontinuous because of the term $-\eta sgn(S)$. This discontinuity may cause chattering, the main drawback of this kind of controller. However, it can be substituted by similar continuous functions such as $\tanh(\kappa S)$ or the sigmoid function, without significant degradation.

According to [12], it is possible to prove that the system is asymptotically stable with the following Lyapunov Function:

$$
V(t) = \frac{1}{2}s^2 \tag{10}
$$

In the next section, this controller is analyzed in more detail for the specific version of SMC applied to the spherical robot system.

2) Hierarchical Sliding Mode Control: HSMC is a variation of SMC designed for under-actuated systems.

Rewriting the system state-space model with additional disturbances:

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= f_1(x) + b_1(x) + d_1(x) \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= f_2(x) + b_2(x) + d_2(x)
\end{aligned}
$$

Two sliding variables are defined:

$$
s_1 = \dot{e}_1 + \lambda_1 e_1 \tag{11}
$$

$$
s_2 = \dot{e}_2 + \lambda_2 e_2 \tag{12}
$$

The associated sliding surfaces ($s_1 = 0$ and $s_2 = 0$) assign desired behaviors for each variable $x_1 = \theta$ and $x_3 = \phi$. Since there is only one input, the HSMC defines a second-level sliding surface:

$$
S = \alpha s_1 + \beta s_2 \tag{13}
$$

with $\alpha > 0$ and $\beta > 0$.

Then, the input is defined as:

$$
u = u_{eq1} + u_{eq2} + u_{sw} \tag{14}
$$

where:

$$
u_{eq1} = -\frac{(f_1(x) + \lambda\dot{e}_1 - \ddot{\theta}_d)}{b_1(x)} \tag{15}
$$

$$
u_{eq2} = -\frac{(f_2(x) + \lambda\dot{e}_2 - \ddot{\phi}_d)}{b_2(x)} \tag{16}
$$

u_{sw} is derived with Lyapunov theory. Similarly to the Lyapunov function 10, the following Lyapunov function is defined:

$$
V(t) = \frac{1}{2}S^2 \tag{17}
$$

Differentiating $V(t)$ with respect to time:

$$
\begin{aligned}
\dot{V}(t) = S\dot{S} &= S[\alpha \boldsymbol{s_1} + \beta \boldsymbol{s_1}] \\
&= S[\alpha(\boldsymbol{\ddot{e}_1} + \lambda_1\dot{e}_1) + \beta(\boldsymbol{\ddot{e}_2} + \lambda_2\dot{e}_2)] \\
&= S[\alpha(f_1 + b_1 u - \ddot{x}_1 + d_1 + \lambda_1\dot{e}_1) + \\
\beta(f_2 + b_2 u - \ddot{x}_2 + d_2 + \lambda_2\dot{e}_2)]
\end{aligned} \tag{18}
$$

Substituting u from eq. 14 into eq. 18:

$$
\begin{aligned}
\dot{V}(t) =& S[\boldsymbol{\alpha b_1(u_{eq1} + u_{sw}) + \beta b_2(u_{eq2} + u_{sw})} + \\
& \alpha d_1 + \beta d_2]
\end{aligned} \tag{19}
$$

Inspired by the controller of the last section, we want to make the bold section of 19 equal to $-\eta sgn(S) - kS$ (with $\eta > 0$ and $k > 0$). Thus:

$$
\alpha b_1(u_{eq1} + u_{sw}) + \beta b_2(u_{eq2} + u_{sw}) = -\eta sgn(S) - kS
$$

$$
u_{sw} = -\frac{(\beta b_2 u_{eq1} + \alpha b_1 u_{eq2} + \eta sgn(S) + kS)}{(\alpha b_1 + \beta b_2)^{-1}} \tag{20}
$$

Substituting u_{eq1} (eq. 15), u_{eq2} (eq.16) and u_{sw} (eq.20) in u (eq. 14):

$$
\boldsymbol{u = \frac{\alpha b_1 u_{eq1} + \beta b_1 u_{eq1} - \eta sgn(S) - kS}{\alpha b_1 + \beta b_1}} \tag{21}
$$

Substituting u_{eq1} (eq. 15), u_{eq2} (eq.16) and u_{sw} (eq.20) in $\dot{V}(t)$ (eq.19):

$$
\dot{V}(t) = S(-\eta sgn(S) - kS + \alpha d_1 + \beta d_2)
$$

Since $Ssgn(S) = |S|$:

$$
\dot{V}(t) = -\eta|S| - kS^2 + S(\alpha d_1 + \beta d_2)
$$

Applying the Cauchy-Schwatz Inequality:

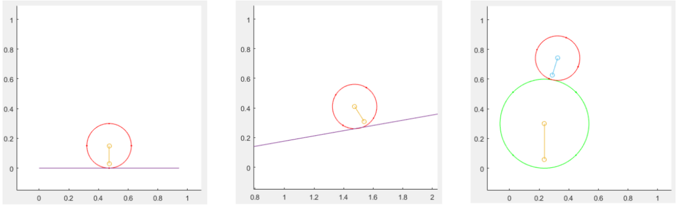
$$
\begin{aligned}
S(\alpha d_1 + \beta d_2) &\leq |S||\alpha d_1 + \beta d_2| \\
\dot{V}(t) &\leq -\eta|S| - kS^2 + |S||\alpha d_1 + \beta d_2| \\
\dot{V}(t) &\leq -kS^2 + |S|(-\eta + |\alpha d_1 + \beta d_2|)
\end{aligned}
$$

Defining $D_M = \sup_{t>0}(|\alpha d_1 + \beta d_2|)$ (upper bound of the disturbance). If $\eta > D_M$, then $\dot{V}(t) < 0$. In addition, notice that $\dot{V}(t) = 0$ if and only if $S = 0$, so by LaSalle's invariance principle, the second-level sliding surface is asymptotically stable [13]. It's also possible to prove that the first-level sliding surfaces s_1 and s_2 are also asymptotically stable. More details can be found in [12].

V. RESULTS

The results of the implementations of each controller under different configurationes are detailed in the sections that follow. Each implementation was simulated in the MATLAB environment using MATLAB's ODE45 solver. MATLAB was also used to generate visual simulations of the results, which are used to verify performance by intuition. See figure 5.

For all the simulations, reasonable parameter values were used, with values based on the parameter values from [9]. In both of the stacked-hoop cases, the bottom hoop values are scaled appropriately from the top hoop values.



Flat Plane

Inclined Plane

Hoop on Hoop

Fig. 5. MATLAB Animation Screenshots

A. Partial Feedback Linearization

This controller did not work for realistic values. The simulation crashes due high torques and velocities for ϕ. Using non-realistic parameters we should see ϕ is unstable. Since the results were terrible, this controller was scrapped.

B. Robust Control

When applied to a single hoop system on a horizontal plane, the Robust Controller produced successful results in tracing the desired trajectory for θ. This can be observed in Fig. 6.

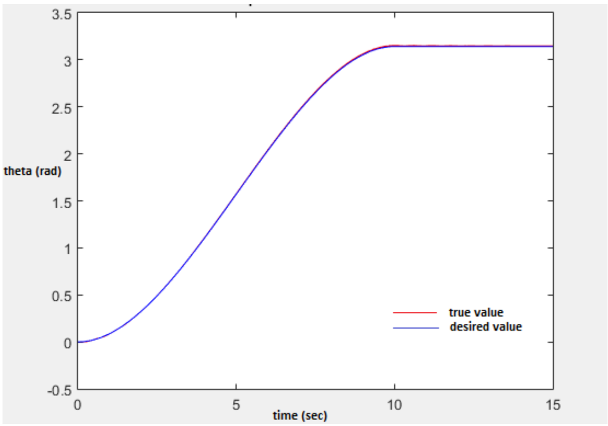


Fig. 6. Robust Controller on a Horizontal Plane

However when this controller was applied to a single hoop system on an inclined plane of $5\deg$, the results were unsuccessful in maintaining the stability of the hoop. As seen in Fig.7, the true values for θ exhibited undamped oscillations; eventually, destabilizing the hoop at the end of the trajectory, and causing it to roll down the slope.

C. Computed Torque Control

1) Hoop on Fixed Hoop: Although computed torque control is intended for fully-actuated systems, it did work well for this system in all the configurations. Results for the flat plane and constant slope are not included to conserve space. Results for the hoop on spherical surface (fixed hoop) are shown in Fig. 8

2) Hoop on Hoop: This technique worked well with proper tuning of the control parameters kp and kv for all the error equations. The results of the test are as shown in Fig. 9:

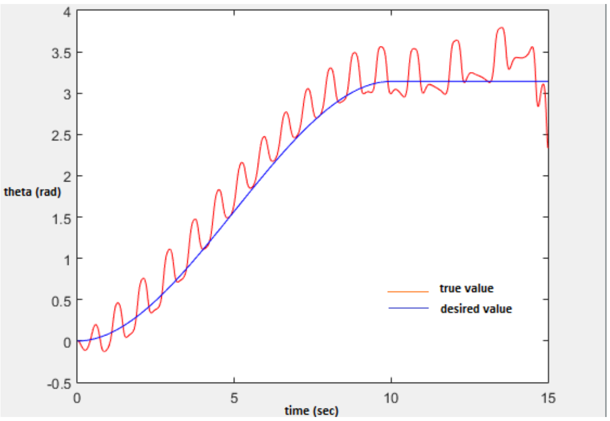


Fig. 7. Robust Controller on an Inclined Plane

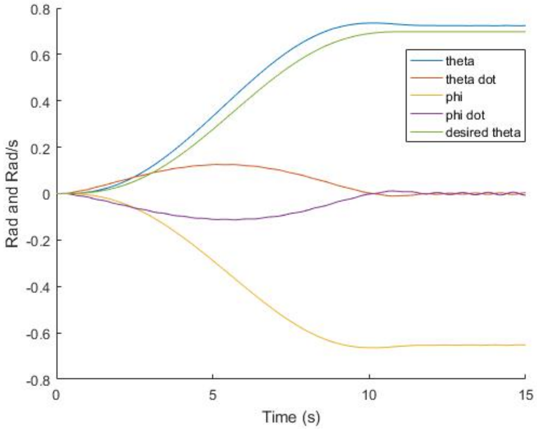


Fig. 8. State Variables for Hoop on Fixed Hoop

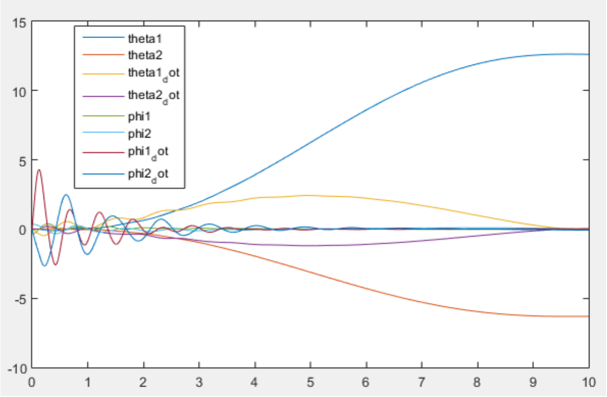


Fig. 9. Hoop-on-Hoop Trajectory following using computed torque. Units: rad and rad/sec on Y axis for angular displacement and velocity state variables. Time in seconds on X axis

The application of computed torque control produces satisfactory results. It was observed that the system follows the desired trajectory smoothly after oscillating a little in the initial span of the simulation. It is notable that the top hoop maintained its balance and remained at the apex position while the bottom hoop translated through a full rotation.

D. Hierarchical Sliding Mode Control

When applied to the single-hoop on an inclined surface with $\theta_d = constant$, $\dot{\theta}_d = \ddot{\theta}_d = 0$; and cubic polynom trajectory for θ; both with $\phi_d = \phi(\gamma)_{equilibrium}$, $\dot{\phi}_d = \ddot{\phi}_d = 0$, the HSMC controller produced good results.

As presented in section IV-D.1, the fact that the input is discontinuous may cause chattering. To mitigate this problem, $sgn(S)$ is replaced with $tanh(\kappa S)$.

The results for the following parameters are presented:

$$\begin{array}{llll} \gamma = 5^\circ & \lambda_1 = 8.0 & \lambda_2 = 25.0 & k = 0.5 \\ \alpha = 12.0 & \beta = 2.0 & \kappa = 40.0 & \eta = 20.0 \end{array}$$

A random disturbance d_1 and d_2 uniformly distributed from 0 to 1, was also added to the system. Notice that $D_M = \sup_{t>0}(|\alpha d_1 + \beta d_2|) = 14.0$ and $\eta = 20.0 > D_M$. The following figures (10, 11, 12) show the the system states compared with the desired state, the input torque, and the sliding variable behavior.

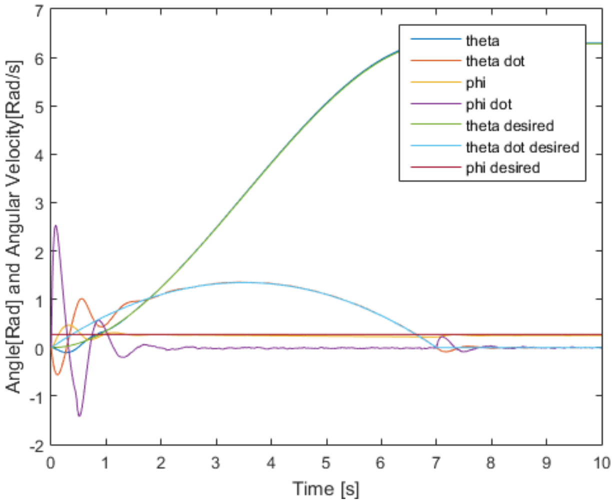


Fig. 10. State Variables for Sliding Mode Control

The drastic change around seven seconds is due the trajectory planning. The desired trajectory until that point is a cubic polynomial function of non-zero final acceleration. After that, the desired trajectory has zero acceleration, causing a discontinuity. However, the system recovered from this disturbance.

Notice also that the torque has relatively low values. It has some noise, but it is still smaller than the original HSMC version with the discontinuous $sgn(S)$ function. All states and sliding surfaces are stable. Notice that S converges very fast followed by s_1 and s_2. However, s_1 and s_2 seem to slightly diverge from each other around two seconds, but they recover at around seven seconds (the discontinuity). This behavior also remains using $sgn(S)$ instead of $\tanh S$ or increasing η. Fortunately, rerunning the simulation with a longer simulation time we could see that it does not diverge, but stabilizes with $|s_1| \neq 0, |s_2| \neq 0$ and $|s_1| < |s_2|$. When the disturbance is removed, this effect basically disappears.

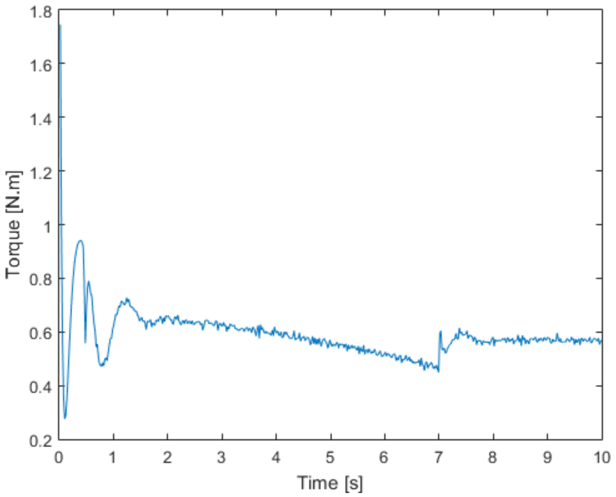


Fig. 11. Input Torque for Sliding Mode Control

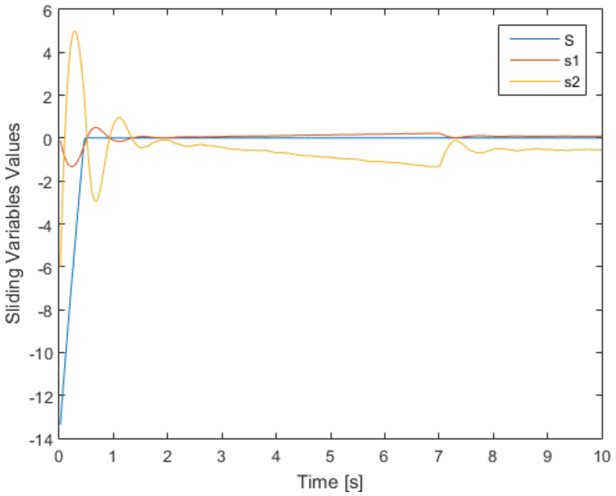


Fig. 12. Sliding Variables

These analysis suggest that this system may not be part of the group of systems described in [12] whose s_1 and s_2 sliding variables are asymptotically stable, at least under situations with high disturbances.

This controller was not tested on the 2-hoop configuration because of time constraints.

VI. Conclusion

This project was successfully completed through the satisfactory completion of both the initial goals and the reach goals. A dynamic model of the spherical robot on an inclined plane was derived from various dynamic models that were found in literature review. The derived model was shown to follow intuitive behavior. Control algorithms that were tested on this model to implement trajectory following include: Computed Torque Control, Robust Control and Hierarchical Sliding Mode Control. Among these, Robust control did not produce desirable results. Moreover, a dynamic model of a stacked-hoop system (one hoop on another) was developed using Lagrangian dynamics. This model was also shown to follow expected behavior when uncontrolled. Computed Torque Control was applied to this model to successfully implement trajectory following of the stacked-hoop system, thus fulfilling the objective.

References

[1] CHASE, Richard; PANDYA, Abhilash. A Review of Active Mechanical Driving Principles of Spherical Robots; Department of Electrical and Computer Engineering, Wayne State University, USA, 2012

[2] CROSSLEY, Vincent A.; A Literature Review on the Design of Spherical Rolling Robots; Department of Mechanical Engineering, Carnegie Mellon University, USA, 2011

[3] “Rotundus,” Rotundus, 2008. [Online]. Available at: http://www.rotundus.se/. [Accessed: 28-Apr-2016].

[4] “Sphero — Connected Entertainment Robots,” Sphero, 2016. [Online]. Available at: http://www.sphero.com/. [Accessed: 28-Apr-2016].

[5] “BB-8,” Wikipedia. [Online]. Available at: https://en.wikipedia.org/wiki/bb-. [Accessed: 28-Apr-2016].

[6] Yu, Tao, et al. ”Stabilization and control of a spherical robot on an inclined plane.” Res. J. Appl. Sci. Eng. Technology 5.6 (2013): 2289-2296.

[7] Shtessel, Yuri, et al. ”Introduction: Intuitive theory of sliding mode control.” Sliding Mode Control and Observation. Springer New York, 2014. 1-42.

[8] Mikhail Svinin, Yang Bai, and Motoji Yamamoto. Dynamic Model and Motion Planning for a Pendulum Actuated Spherical Rolling Robot . Washington State Convention Center, 2015

[9] Tao Yu and Hanxu Sun. “Variable Structure Control of Spherical Robots with Exponential Reaching Law”. Studies in System Science (SSS) Volume 2. 2014.

[10] Zhao, Bo, et al. ”Dynamics and motion control of a two pendulums driven spherical robot.” Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. IEEE, 2010.

[11] Bai, Yang, Mikhail Svinin, and Motoji Yamamoto. ”Motion planning for a pendulum-driven rolling robot tracing spherical contact curves.” Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. IEEE, 2015.

[12] Wang, W., Yi, J., and Zhao D. “Design of Sliding mode Controller Based on Stable Analysis for a Class of Under actuated Systems.” Information and Control, 2005.

[13] Yue, Ming, and Baoyin Liu. ”Adaptive control of an underactuated spherical robot with a dynamic stable equilibrium point using hierarchical sliding mode approach.” International Journal of Adaptive Control and Signal Processing 28.6 (2014): 523-535.

[14] Edwards, Christopher, Leonid Fridman, and Arie Levant. Sliding mode control and observation. New York, NY, USA: Birkhäuser, 2014.

[15] M. W. Spong,” On the robust control of robotic manipulators,” *IEEE Transactions on Automatic Control*, vol. 37, no. 11, pp. 1782 - 1786, 1992