Omnidirectional Locomotion Control of a Pendulum Driven Spherical Robot

Narendran Muraleedharan
Embry-Riddle Aeronautical University
Prescott, AZ 86301
muraleen@my.erau.edu

Daniel S. Cohen
Embry-Riddle Aeronautical University
Prescott, AZ 86301
cohend1@my.erau.edu

Douglas R. Isenberg
Embry-Riddle Aeronautical University
Prescott, AZ 86301
isenberd@erau.edu

***Abstract*—This paper proposes a method for omnidirectional locomotion control of a pendulum driven spherical robot. Two continuous rotation actuators control the orientation of a pendulum within the spherical shell. The control system generates desired actuator angles to move the pendulum in the direction of the desired motion in reference to an assumed inertial frame. Additionally, partial feedback linearizion of the under-actuated nonlinear system aids in driving the actuator angles to the desired angles. The dynamics of the robot were derived using the Euler-Lagrange formulation and simulated to verify the functionality of the controller. The simulations show the proposed control method is capable of performing directional and set-point control of the robot.**

I. Introduction

Spherical mobile robots have an external spherical shell which use various internal mechanisms as opposed to wheels for locomotion. Spherical mobile robots also have the ability to perform omnidirectional locomotion as opposed to four-wheeled robots - which for the purposes of this paper, is assumed to be the ability to move in any direction from any position and orientation. Spherical robots can perform omnidirectional locomotion by rolling in the desired direction.

A significant amount of research has been conducted over the past decade on design of spherical robot actuation mechanisms and control systems. The most commonly employed internal actuation mechanisms for spherical robots are the barycenter offset system and an internally actuated pendulum. The barycenter offset system uses a wheeled robot inside a spherical shell. By moving the wheeled robot, the center of gravity of the system shifts and induces a torque on the outer shell. This method can inhibit the robot from being able to accelerate in any desired direction while moving. Previous research on such robots show the viability of the mechanism for linear motion and driving in paths with small curvature. However, such systems are shown to exhibit trouble driving in paths with large curvatures [1], [2]. A pendulum driven robot by contrast, uses an internal ballast which can be moved around inside the shell. The pendulum also induces a torque on the outer shell and allows for locomotion. Other spherical robot actuation mechanisms include an internal drive unit, universal wheel, double pendulum, control moment gyroscopes (CMG), shell transformation systems etc [3], [4].

978-1-5090-2246-5/16/$31.00

This paper considers the use of a pendulum driven spherical robot with two orthogonally placed actuators to move the ballast inside the shell. Pendulum driven spherical robots have been studied extensively and various control systems have been developed. A common approach to controlling pendulum driven robots is by controlling the drive speed of the robot with one of the actuators and the steering with the other [5]–[9]. A drive and steer based control system inhibits the robot's ability to perform omnidirectional locomotion. Usually, such robots can be directly teleoperated and do not need any automatic control [6]. However, controllers can be developed to stabilize the drive speed and lean angle for steering [7].

Some interesting alternative controllers applied to pendulum driven spherical robots include the use of various neural network structures and exponential reaching law (ERL) controllers. The study on the neural network controller by Cai et al, provides promising results for linear motion, however it does not address two dimensional (2D) control of the robot [5]. Yu and Sun's study on the sliding mode approach for an ERL controller attains omnidirectional locomotion control of a pendulum driven spherical robot (named BYQ-VIII) on an inclined plane. The method has been tested in simulation and experimentally and proven to be functional [10]. The method proposed in this paper approaches the problem from a simpler perspective using some reasonable assumptions for low speed omnidirectional locomotion.

The proposed controller is comprised of two stages. The first stage is used to transform task space command such as target waypoints or velocity vectors to joint space and does not require any knowledge of the system dynamics. The second stage uses partial feedback linearization to control the actuator angles but can be replaced with a controller without knowledge of the dynamic parameters of the robot. The use of well controlled servo actuators can also eliminate the need for the dynamic parameters.

II. Dynamic Modeling

The dynamical model of the spherical robot is described with four coordinate frames - I indicating the inertial frame, S indicating the sphere body frame fixed at the center of the robot, D indicating the drive spar frame fixed at the center of the first rotating link and P indicating the pendulum frame

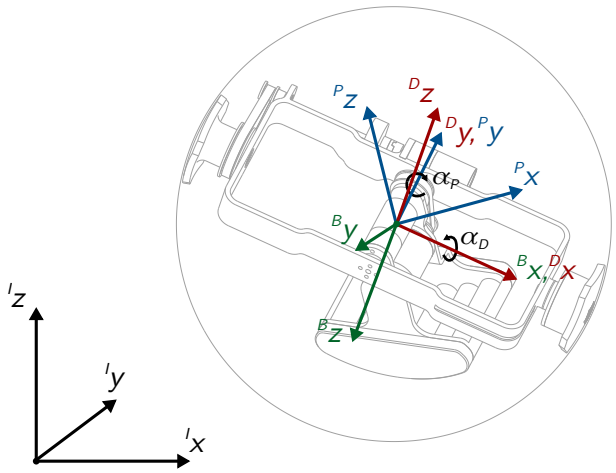

Fig. 1. Figure of a pendulum driven spherical mobile robot showing the coordinate frames employed.

with its origin at coincidental with the D and S frames but able to rotate with the second rotating link.

The state vector of the system $\boldsymbol{\gamma}$ is defined as:

$$\boldsymbol{\gamma} = \left[x, y, \varphi, \vartheta, \psi, \alpha_D, \alpha_P\right]^T, \tag{1}$$

where x and y describe the position of the robot's sphere frame S with respect to the inertial axes x^I and y^I respectively. The angles φ, ϑ and ψ describe an XYZ Euler rotation sequence for the transformation between the inertial frame and the sphere frame. The actuator angles for the drive spar and pendulum are represented by α_D and α_P respectively.

The sphere's body frame S, drive spar frame D and the pendulum frame P share the same origin as shown in figure 1. The transformation matrix from the inertial frame I to the pendulum frame P is expressed as shown in equation (2).

$$\begin{aligned} {}^I\boldsymbol{T}_P &= {}^I\boldsymbol{T}_S {}^S\boldsymbol{T}_D {}^D\boldsymbol{T}_P \\ &= \boldsymbol{R}_{x,\varphi}\boldsymbol{R}_{y,\vartheta}\boldsymbol{R}_{z,\psi}\boldsymbol{R}_{x,\alpha_D}\boldsymbol{R}_{y,\alpha_P}. \end{aligned} \tag{2}$$

In this paper, rotation matrices are represented in the forms $\boldsymbol{R}_{x,\zeta}$, $\boldsymbol{R}_{y,\zeta}$ and $\boldsymbol{R}_{z,\zeta}$ where the matrices describe rotations about the relative x, y and z axes respectively by the angle ζ.

In equation (2), ${}^I\boldsymbol{T}_S$ is the transformation from the inertial frame I to the frame S which is described by the set of XYZ Euler angle rotations mentioned earlier. ${}^S\boldsymbol{T}_D$ is the transformation from S frame to frame D describe by rotation matrix $\boldsymbol{R}_{x,\alpha_D}$, and ${}^D\boldsymbol{T}_P$ is the transformation from frame D to the frame P described by the rotation matrix $\boldsymbol{R}_{y,\alpha_P}$.

The Euler-Lagrange formulation with a kinematic constraint is used to derive the equations of motion of the system as shown in equation (3).

$$\left(\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\boldsymbol{\gamma}}}\right) - \frac{\partial L}{\partial \boldsymbol{\gamma}} + \boldsymbol{\lambda}\boldsymbol{A}(\boldsymbol{\gamma})\right)^T = \boldsymbol{f}, \tag{3}$$

where $\boldsymbol{f}$ is the vector of forces along the state vector coordinates, L is the Lagrangian of the system described by $L = K - U$, $\boldsymbol{\lambda}$ is a Lagrange multiplier, and $\boldsymbol{A}(\boldsymbol{\gamma})$ is the Pfaffian constraint matrix. The kinetic energy of the system is calculated using the equation:

$$\begin{aligned} K = \sum \Bigg(&\frac{1}{2} m_B {}^I_I\dot{\boldsymbol{r}}^T_B {}^I_I\dot{\boldsymbol{r}}_B + m_B {}^I_I\dot{\boldsymbol{r}}^T_B {}^I\dot{\boldsymbol{T}}_B {}^B_B\boldsymbol{r}_{cm_B} \\ &+ \frac{1}{2}\text{trace}({}^I\dot{\boldsymbol{T}}_B {}^B_B\hat{\boldsymbol{J}} {}^I\dot{\boldsymbol{T}}^T_B)\Bigg), \end{aligned} \tag{4}$$

where m_B is the mass of body B, ${}^I_I\boldsymbol{r}_B$ is the position of the body frame B with respect to the inertial frame I, ${}^I\boldsymbol{T}_B$ is the transformation from frame I to B, ${}^B_B\boldsymbol{r}_{cm_B}$ is the position vector of the center of mass of the frame B measured with respect to frame B and ${}^B_B\hat{\boldsymbol{J}}$ is the pseudo-inertia tensor of the body $\forall B \in \{S, D, P\}$.

The pseudo-inertia tensor ${}^B_B\hat{\boldsymbol{J}}$ is calculated from the inertia tensor ${}^B_B\boldsymbol{J}$ as shown in equation (5).

$${}^B_B\hat{\boldsymbol{J}} = \frac{1}{2}\text{trace}({}^B_B\boldsymbol{J})I - {}^B_B\boldsymbol{J}. \tag{5}$$

The potential energy of the system is calculated using the equation:

$$U = \sum [0, 0, g_z]({}^I_I\boldsymbol{r}_B + {}^I\boldsymbol{T}_B {}^B_B\boldsymbol{r}_{cm_B}) m_B, \tag{6}$$

where g_z is the gravitational acceleration acing over the body in the inertial z direction.

It is well known that the Euler-Lagrange equation can be expressed in the form:

$$\boldsymbol{f} = \boldsymbol{H}(\boldsymbol{\gamma})\ddot{\boldsymbol{\gamma}} + \boldsymbol{d}(\boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}) + \boldsymbol{g}(\boldsymbol{\gamma}) + \boldsymbol{A}(\boldsymbol{\gamma})^T\boldsymbol{\lambda}, \tag{7}$$

where $\boldsymbol{H}(\boldsymbol{\gamma})\ddot{\boldsymbol{\gamma}}$ is the system mass matrix, $\boldsymbol{d}(\boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}})$ is the vector of Coriolis, centripetal and frictional forces and torques and $\boldsymbol{g}(\boldsymbol{\gamma})$ is the vector gravitational forces and torques. Viscous and Coulombic friction models are used for the actuator angles.

The Pfaffian constraint given by the equation:

$$\boldsymbol{A}(\boldsymbol{\gamma})\dot{\boldsymbol{\gamma}} = \boldsymbol{0}, \tag{8}$$

simulates a no-slip condition in the inertial x and y axes. The constraint matrix $\boldsymbol{A}(\boldsymbol{\gamma})$ is derived using the relationship:

$${}^I_I\dot{\boldsymbol{r}}_S = [\dot{x}, \dot{y}, 0]^T = r({}^I_S\boldsymbol{\omega}^I \times {}^I\boldsymbol{z}), \tag{9}$$

where r is the radius of the sphere, ${}^I\boldsymbol{z}$ is a unit vector in the inertial z-axis, and ${}^I_S\boldsymbol{\omega}^I$ is the angular velocity vector of the sphere which is calculated using equation (10).

$$Sk({}^I_S\boldsymbol{\omega}^I) = {}^I\dot{\boldsymbol{T}}_S {}^I\boldsymbol{T}^T_S, \tag{10}$$

where $Sk({}^I_S\boldsymbol{\omega}^I)$ is a skew-symmetric matrix with the components of vector ${}^I_S\boldsymbol{\omega}^I$ in the off-diagonals.

Equation (11) results in a constraint matrix $\boldsymbol{A}(\boldsymbol{\gamma})$ of:

$$\boldsymbol{A}(\boldsymbol{\gamma}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & r \\ -r\cos(\varphi) & 0 \\ r\cos(\vartheta)\sin(\varphi) & \sin(\vartheta) \\ 0 & 0 \\ 0 & 0 \end{bmatrix}^T. \quad (11)$$

The system's equations of motion described in equation (7) is combined with the derivative of the constraint equation (8) to form the matrix equation:

$$\begin{bmatrix} \boldsymbol{H}(\boldsymbol{\gamma}) & \boldsymbol{A}(\boldsymbol{\gamma})^T \\ \boldsymbol{A}(\boldsymbol{\gamma}) & \boldsymbol{0}_{2\times 2} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\gamma}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} - \boldsymbol{d}(\boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}) - \boldsymbol{g}(\boldsymbol{\gamma}) \\ -\dot{\boldsymbol{A}}(\boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}})\dot{\boldsymbol{\gamma}} \end{bmatrix}. \quad (12)$$

The state vector accelerations $\ddot{\boldsymbol{\gamma}}$ is solved for from equation 11, and then integrated using a fourth-order Runge-Kutta (RK4) to simulate the dynamics of the robot.

III. Control System Design

The control system is divided into two stages. In the first stage, task space commands are transformed into the joint space. In the second stage, the joint space angles calculated by the first stage are utilized to drive the actuator angles through the use of partial feedback linearization control law.

A. Task-space to Joint-space transformation

The proposed control system assumes that the robot accelerates in the direction the pendulum is lifted. Additionally, a no slip condition is assumed between the sphere and the ground, and the robot is simplified into a two-body system comprised of the sphere and the pendulum. Furthermore, it is assumed that the state velocities are small so that the vector of Coriolis and centripetal forces $\boldsymbol{d}$ becomes negligible.

The magnitude of acceleration can be controlled by the angle between the pendulum and the vertical, β_1, while the direction of acceleration can be controller by the rotation of the pendulum about the inertial z-axis, β_1. The angles β_1 and β_2 are depicted in figure 2.

By applying force equilibrium about the driving direction:

$$\Sigma F_y = F_f = (m_p + m_s)\ddot{y}, \quad (13)$$

and moment equilibrium equations about the center of the sphere for a one-dimensional (1D) case:

$$\Sigma M_x = F_f r - m_p g d \sin(\beta_1) = J\ddot{\varphi} \quad (14)$$

and the kinematic constraint, the required angle between the pendulum and the vertical β_1 can be computed.

In equations (13) and (14), F_f is the force of friction at the point of contact between the spherical shell and the ground, r is the radius of the sphere, m_p is the mass of the pendulum ballast, m_s is the mass of the sphere, d is the distance between the center of the sphere and the center of mass of the pendulum, φ is the angle of rotation in the 1D system and y is the linear translation in the 1D system.

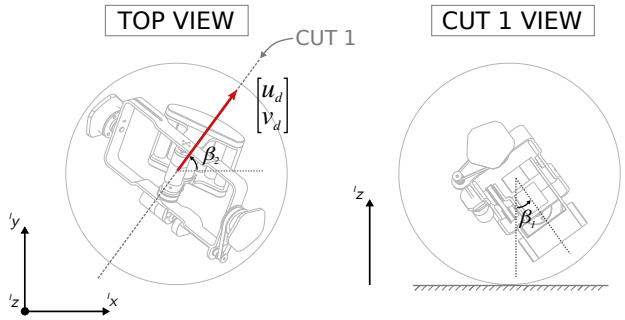

Fig. 2. Top and sectional view of the spherical robot to show angles β_1 and β_2.

The kinematic Pfaffian constraint discussed earlier can be written in the 1D system as:

$$\dot{y} = r\dot{\varphi} \quad \Longrightarrow \quad \ddot{y} = r\ddot{\varphi}. \quad (15)$$

By substituting F_f from equation (13) and $\ddot{\varphi}$ from equation (15) into equation (14):

$$\beta_1 = \sin^{-1}\left(\left(\left(1 + \frac{m_s}{m_p}\right)r - \frac{J}{rm_p}\right)\frac{\ddot{y}}{gd}\right), \quad (16)$$

where J is the rotational inertia of the system about the rolling direction. For small angles of β_1, a proportional relationship between β_1 and $\ddot{y}$ is assumed. The target acceleration components to meet a given desired velocity vector are calculated using PD controllers.

The angle β_1 is then computed as the magnitude of the target acceleration components as shown below:

$$\beta_1 = \sqrt{u_d^2 + v_d^2}, \quad (17)$$

where u_d and v_d are the x and y components respectively of the desired acceleration in the inertial frame.

The direction of the desired acceleration vector is computed using equation 18:

$$\beta_2 = \text{atan2}(v_d, u_d). \quad (18)$$

The trajectory tracking controller computes the target acceleration components as shown in equation (19):

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = \begin{bmatrix} \ddot{x}_d \\ \ddot{y}_d \end{bmatrix} + K_p\left(\begin{bmatrix} x_d \\ y_d \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix}\right) + K_d\left(\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} - \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}\right), \quad (19)$$

where K_p and K_d are the proportional and derivative gains which are calculated from a desired damping ratio and natural frequency.

Once the desired β_1 and β_2 angles are computed, the required actuator angles α_D and α_P need to be computed to move the pendulum to the desired position.

The transformation matrix from the inertial frame I to the pendulum frame P is described both by equation (20) and

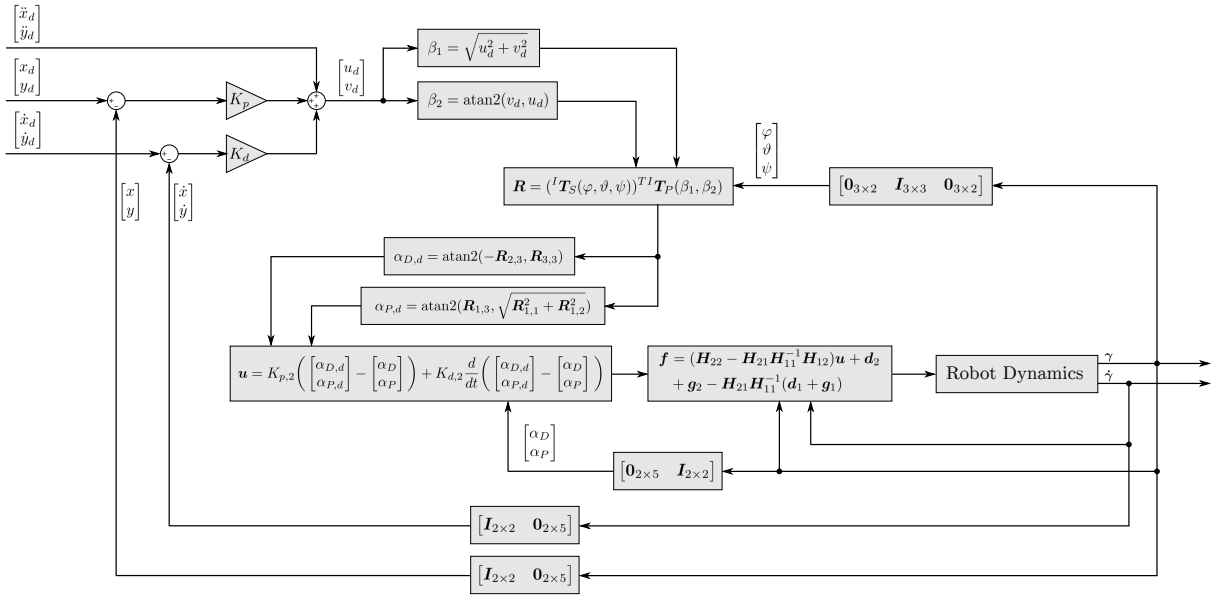

Fig. 3. Proposed control method block diagram.

the product of ${}^I\boldsymbol{T}_S$ and ${}^S\boldsymbol{T}_P$ from equations (21) and (22) as shown in equation (23):

$$ {}^I\boldsymbol{T}_P(\beta_1,\beta_2) = \boldsymbol{R}_{z,\beta_2}\boldsymbol{R}_{x,\beta_1}, \tag{20}$$

$$ {}^I\boldsymbol{T}_S(\varphi,\vartheta,\psi) = \boldsymbol{R}_{x,\varphi}\boldsymbol{R}_{y,\vartheta}\boldsymbol{R}_{z,\psi}, \tag{21}$$

$$ {}^S\boldsymbol{T}_P(\alpha_D,\alpha_P) = \boldsymbol{R}_{x,\alpha_D}\boldsymbol{R}_{y,\alpha_P}, \tag{22}$$

$$ {}^I\boldsymbol{T}_P(\beta_1,\beta_2) = {}^I\boldsymbol{T}_S(\varphi,\vartheta,\psi){}^S\boldsymbol{T}_P(\alpha_D,\alpha_P). \tag{23}$$

This allows the transformation matrix composed of the actuator angles ${}^S\boldsymbol{T}_P(\alpha_D,\alpha_P)$ to be calculated using the equation:

$$ {}^S\boldsymbol{T}_P(\alpha_D,\alpha_P) = ({}^I\boldsymbol{T}_S(\varphi,\vartheta,\psi))^{-1}{}^I\boldsymbol{T}_P(\beta_1,\beta_2). \tag{24}$$

As the transformation matrices are orthogonal, the transpose of the matrix could be exploited instead of inverse, as shown in equation (25):

$$ {}^S\boldsymbol{T}_P(\alpha_D,\alpha_P) = ({}^I\boldsymbol{T}_S(\varphi,\vartheta,\psi))^{TI}\boldsymbol{T}_P(\beta_1,\beta_2). \tag{25}$$

The left hand side of equation (25) is a two axis rotation, whereas the right hand side can be any orthogonal matrix. This requires a *virtual* rotation matrix to be multiplied to the left hand side to allow for complete three rotation. A rotation matrix about the z-axis by a *virtual* angle a_v is utilized as shown in equation (26):

$$ \boldsymbol{R} = {}^S\boldsymbol{T}_P(\alpha_D,\alpha_P) = \boldsymbol{R}_{x,\alpha_D}\boldsymbol{R}_{y,\alpha_P}R_{z,\alpha_v}. \tag{26}$$

The desired actuator angles $\alpha_{1,d}$ and $\alpha_{2,d}$ are extracted from the rotation matrix $\boldsymbol{R}$ using equations (27) and (28):

$$ \alpha_{1,d} = \text{atan2}(-\boldsymbol{R}_{2,3}, \boldsymbol{R}_{3,3}), \tag{27}$$

$$ \alpha_{2,d} = \text{atan2}(\boldsymbol{R}_{1,3}, \sqrt{\boldsymbol{R}_{1,1}^2 + \boldsymbol{R}_{1,2}^2}), \tag{28}$$

where $\boldsymbol{R}_{i,j}$ is the scalar in the i^{th} row and j^{th} column of the matrix $\boldsymbol{R}$.

The equations are selected so the virtual angle α_v does not affect the actuator angles.

B. Partial Feedback Linearization

The system's equations of motion are separated into the unactuated and actuated equations as shown in equation (29).

$$ \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{f}\end{bmatrix} = \begin{bmatrix}\boldsymbol{H}_{11} & \boldsymbol{H}_{12}\\ \boldsymbol{H}_{21} & \boldsymbol{H}_{22}\end{bmatrix}\begin{bmatrix}\ddot{\boldsymbol{\gamma}}_1\\ \ddot{\boldsymbol{\gamma}}_2\end{bmatrix} + \begin{bmatrix}\boldsymbol{d}_1\\ \boldsymbol{d}_2\end{bmatrix} + \begin{bmatrix}\boldsymbol{g}_1\\ \boldsymbol{g}_2\end{bmatrix}, \tag{29}$$

where the vector of forces $\boldsymbol{f} = [F_1, F_2]^T$, the unactuated state vector accelerations $\ddot{\boldsymbol{\gamma}}_1 = [\ddot{x}, \ddot{y}, \ddot{\varphi}, \ddot{\vartheta}, \ddot{\psi}]^T$, the actuated state vector accelerations $\ddot{\boldsymbol{\gamma}}_2 = [\ddot{\alpha}_D, \ddot{\alpha}_P]^T$, $\boldsymbol{H}_{11}$, $\boldsymbol{H}_{12}$, $\boldsymbol{H}_{21}$ and $\boldsymbol{H}_{22}$ are the components of the system mass matrix $\boldsymbol{H}$, $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ are the unactuated and actuated components of the vector of Coriolis, centripetal and frictional forces $\boldsymbol{d}$ and $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are the unactuated and actuated components of the gravitational force vector $\boldsymbol{g}$.

The matrix equation (29) can be written as the unactuated system equation (30):

$$ \boldsymbol{0} = \boldsymbol{H}_{11}\ddot{\boldsymbol{\gamma}}_1 + \boldsymbol{H}_{12}\ddot{\boldsymbol{\gamma}}_2 + \boldsymbol{d}_1 + \boldsymbol{g}_1, \tag{30}$$

and the actuated system equation (31):

$$ \boldsymbol{f} = \boldsymbol{H}_{21}\ddot{\boldsymbol{\gamma}}_1 + \boldsymbol{H}_{22}\ddot{\boldsymbol{\gamma}}_2 + \boldsymbol{d}_2 + \boldsymbol{g}_2. \tag{31}$$

Equation (30) is solved for $\ddot{\boldsymbol{\gamma}}_1$ and substituted into equation (31) to get equation (32):

$$ \begin{aligned}\boldsymbol{f} = {} & (\boldsymbol{H}_{22} - \boldsymbol{H}_{21}\boldsymbol{H}_{11}^{-1}\boldsymbol{H}_{12})\ddot{\boldsymbol{\gamma}}_2 + \boldsymbol{d}_2 + \boldsymbol{g}_2\\ & - \boldsymbol{H}_{21}\boldsymbol{H}_{11}^{-1}(\boldsymbol{d}_1 + \boldsymbol{g}_1).\end{aligned} \tag{32}$$

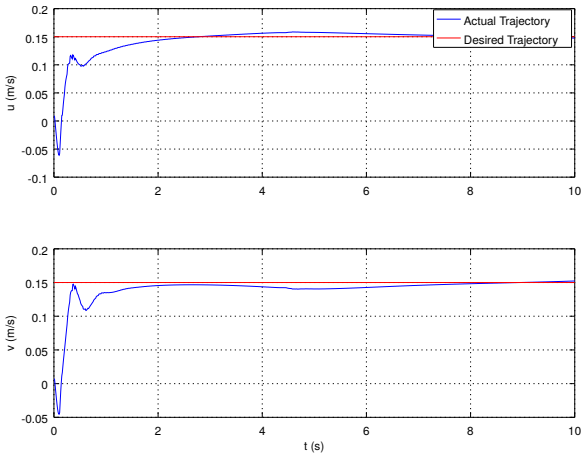

Fig. 4. Plot of velocity components with the robot running the direction controller.

The actuator dynamics of the system can be incorporated into equation (32) to obtain a vector of control inputs such as armature voltages instead of the applied forces vector $\boldsymbol{f}$.

The actuated state vector accelerations $\ddot{\boldsymbol{\gamma}}_2$ are substituted with an input vector $\boldsymbol{u}$ which is computed using a PD controller:

$$\boldsymbol{u} = K_{p,2}\left(\begin{bmatrix}\alpha_{D,d}\\ \alpha_{P,d}\end{bmatrix} - \begin{bmatrix}\alpha_D\\ \alpha_P\end{bmatrix}\right) + K_{d,2}\frac{d}{dt}\left(\begin{bmatrix}\alpha_{D,d}\\ \alpha_{P,d}\end{bmatrix} - \begin{bmatrix}\alpha_D\\ \alpha_P\end{bmatrix}\right), \tag{33}$$

where the gains $K_{p,2}$ and $K_{d,2}$ are set for a desired natural frequency and damping ratio.

The complete block diagram of the control system is illustrated in Figure 3.

IV. Simulation Results

The equations of motion derived in the dynamic modeling section were integrated using an RK4 integrator to simulate the robot and controller. The direction control functionality was tested by commanding a certain desired velocity vector within the designed operational range and plotting the achieved velocity vector components. The control system was also tested with a desired trajectory input and compared against the simulated robot trajectory.

A. Direction Control

The controller's ability to drive at desired velocity vectors was tested by applying a step input to the desired velocity. x and y velocities of $0.15m/s$ were commanded from the controller and PD controllers were used to compute u_d and v_d. Figure 4 shows plots of the actual and desired velocity components of the robot with direction control.

Figure 4 shows the robot's ability to meet a desired velocity vector. Further tuning of the PD controller gains improves the controller response. The errors after initial settling can be attributed to the inertial effects on the dynamics which were assumed to be negligible for controller design. However, these errors are corrected by the integral component of the controller.

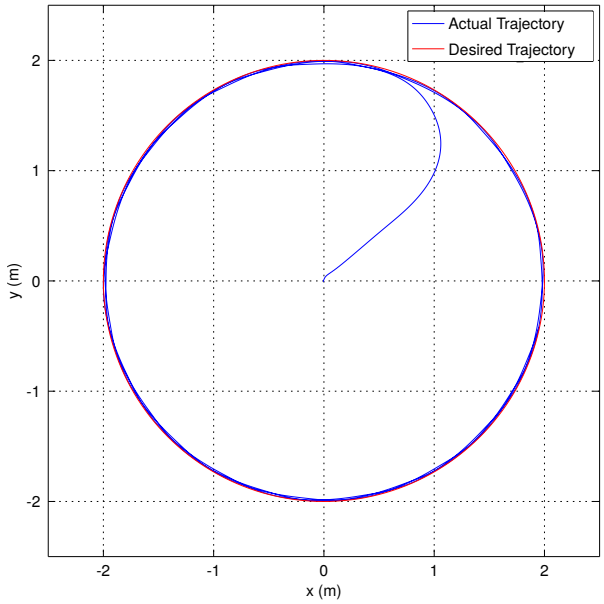

Fig. 5. Trajectory plot of robot running the trajectory tracking controller.

B. Trajectory Tracking Control

The trajectory tracking controller uses equation (19) to compute the target acceleration vector components u_d and v_d. A desired trajectory in the form of desired position, velocity and accelerations are commanded. For the test scenario, a circular trajectory of radius $2m$ and a frequency of $0.05Hz$ was commanded.

Figure 5 is a trajectory plot (y vs. x) of the robot when tracking the circular trajectory. As seen in the plot, the robot starts from the origin and tracks the commanded circular trajectory and is able to maintain the desired trajectory. The individual position and velocity components of the robot are also plotted in figure 6.

The robot however uses two-axis gimbal-like internal mechanism which brings about the possibility of the mechanism entering a gimbal lock in two dimensions. The gimbal lock occurs when the actuator angle α_2 is $\pi/2+n\pi$ where n is any integer. This results in very high actuator commands when α_2 nears the gimbal lock condition and can cause the output to be saturated. A possible solution is the addition of a potential field based term in the controller which pushes the mechanism away from the gimbal lock condition. This prevents the pendulum from passing through the drive spar and inhibits the robot from rolling over about the body y-axis. However as seen in figure 7, when tracking the desired circular trajectory, α_2 stays within a $\pm\pi/2$ range.

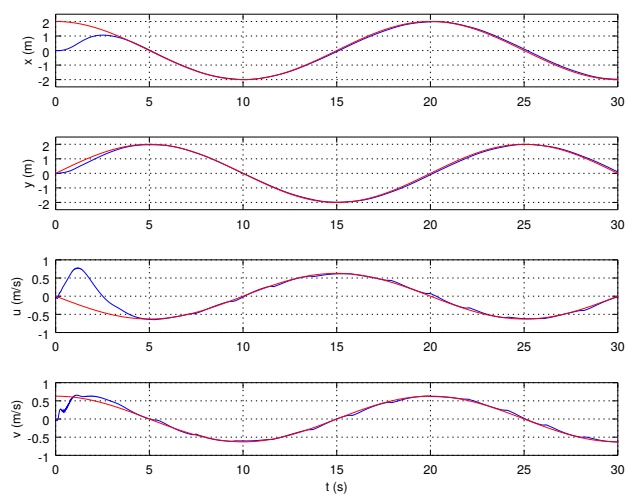

Fig. 6. Plot of position and velocity components with the robot running the trajectory tracking controller.

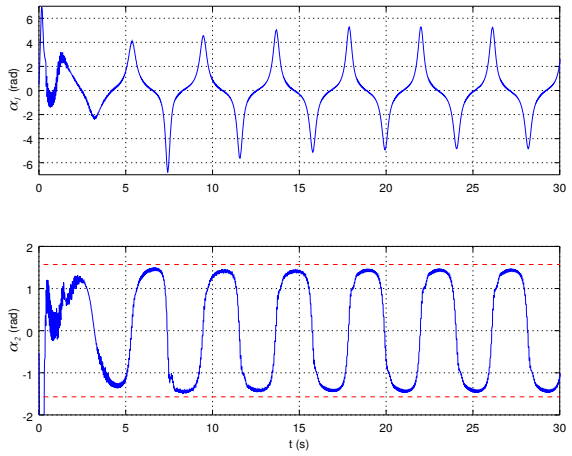

Fig. 7. Plot of actuator angles α_1 and α_2 with the robot running the trajectory tracking controller.

V. Conclusion

The simulations results show that the control method for pendulum driven spherical robots presented in this paper is able to drive the robot to a commanded velocity vector or track a desired trajectory consistently. The first stage of the controller is utilized to transform a given task space (inertial frame) command into joint space and does not require knowledge of the robot's dynamics parameters. The second stage of the controller utilizes partial feedback linearization to drive the actuator angles to the target angles calculated by the first stage. The second stage requires the knowledge of the robot's dynamics parameters, but can be replaced with a simpler controller. The robot can also be driven with continuous rotation servos where the internal servo controllers can be used as a substitute for the second stage.

A. Future Development

The research team is currently constructing a pendulum driven spherical robot to perform experimental validation of the controller. The robot uses two actuators - the first to rotate the drive spar and the pendulum about the sphere's x-axis and the second to rotate the pendulum about the drive spar's y-axis as used in the dynamical model presented in this paper.

The proposed controller will be programmed into the on-board computer of the spherical robot and various trajectories and commanded velocities will be tested.

Acknowledgements

This research is part of a senior capstone project at Embry-Riddle Aeronautical University - Prescott campus. The complete design team includes Gabe Bentz, John Cybulski and Micaela Stewart. The authors would like to thank Dr. Iacopo Gentilini and Dr. Stephen Bruder and the Embry-Riddle development fund for their help and support.

References

[1] J. Alves and J. Dias, "Design and control of a spherical mobile robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 217, no. 6, pp. 457–467, 2003.

[2] N. Xuelei, A. P. Suherlan, G. S. Soh, S. Foong, K. Wood, and K. Otto, "Mechanical development and control of a miniature spherical rolling robot," in *The 13th International Conference on Control, Automation, Robotics and Vision 2014*, 2014.

[3] R. Chase and A. Pandya, "A review of active mechanical driving principles of spherical robots," *Robotics*, vol. 1, no. 1, pp. 3–23, 2012.

[4] T. Ylikorpi and J. Suomela, *Ball-shaped robots*. INTECH Open Access Publisher, 2007.

[5] Y. Cai, Q. Zhan, and X. Xi, "Neural network control for the linear motion of a spherical mobile robot," *International Journal of Advanced Robotic Systems*, vol. 8, no. 4, pp. 79–87, 2011.

[6] A. Halme, T. Schönberg, and Y. Wang, "Motion control of a spherical mobile robot," in *Advanced Motion Control, 1996. AMC'96-MIE. Proceedings., 1996 4th International Workshop on*, vol. 1. IEEE, 1996, pp. 259–264.

[7] D. Liu, H. Sun, Q. Jia, and L. Wang, "Motion control of a spherical mobile robot by feedback linearization," in *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*. IEEE, 2008, pp. 965–970.

[8] M. Nagai, "Control system for a spherical robot," 2008.

[9] K. Othman, "Dynamics control of pendulums driven spherical robot," in *Applied Mechanics and Materials*, vol. 315. Trans Tech Publ, 2013, pp. 192–195.

[10] T. Yu and H. Sun, "Point-to-point motion control for a spherical robot on an inclined plane," *Studies in System Science*, vol. 2, 2014.