École Centrale de Nantes

M1 - Control and Robotics

Dynamic Modelling of an Energy Efficient Rolling Robot

Authors:
Ashwin Bose
Sooraj Anilkumar

Supervisors:
Rafael Balderas Hill
Sébastien Briot

July 7, 2018

Abstract

In recent years, the study of rolling robots has attracted profuse interest for their holonomic characteristics. Their ability to encompass all the hardware components within the spherical shell makes them ideal for hazardous environments. One of the major restrictions that prevent the rolling robots (in fact any mobile robot) in being deployed for real world applications is its operation time. In this project, we explore the applicability of variable stiffness actuators (VSAs) in improving the energy efficiency of locomotion. The underlying principle is to periodically store and release the potential energy in the elastic elements of the actuator, thereby reducing the torque requirement.

This project report presents an analytical model of a reciprocating mass based spherical rolling robot with variable stiffness actuators. The developed analytical model is validated with the help of ADAMS, the standard tool for multibody dynamic simulations.

Subsequently, a simple algorithm to control the motion of the rolling robot is proposed. The results from the closed loop control using ADAMS/Simulink co-simulation demonstrate the effectiveness of the control strategy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The classical approach to increase the autonomy of a mobile robot is to increase the power supply capacity by the addition of more batteries and electrical components. The increase in operation time of the robot is marginal as the added weight due to the additional power supply components will in turn lead to an increase in power demand, thus failing its essential purpose.

An alternative method to increase the energy efficiency of robots, using variable stiffness actuators (VSAs), was introduced by Rafael Balderas et al. for a high-speed pick-and-place robot [1], and by M. Furet [2] for a 2D rolling robot with mobile masses. The concept proposed in the two works is to use a variable stiffness springs (VSS) in parallel to the motors and establish the adaptation of the VSS in such a way to minimize the input torques [3], consequently decreasing the energy consumption of the robot.

The aim of the project is to extend the work done by M. Furet [2] on a 2D rolling robot with mobile masses to a 3D model with mobile masses and VSS. The focus of the project is to compute and validate the dynamic model of a 3D rolling robot with mobile masses in the first phase and then with VSAs, so that the model can be employed in the research on minimizing input torques by Rafael Balderas.

The report is organized as follows: Chapter 2, gives a literature review on the different types of rolling robots in existence and on variable stiffness actuators. The analytical model of the three rolling robots considered: without elastic elements (Nominal), with fixed stiffness springs in parallel and with VSS in parallel, is derived in Chapter 3. Chapter 4 deals with validation of the dynamic models and discussion of the results. In Chapter 5, different control and trajectory generation algorithms are explored to drive the rolling robot. Finally, in the last chapter, conclusions are drawn and proposals for future work are given.

Chapter 2

Literature Review

Mobile robots are ideal for a diverse set of tasks ranging from indoor applications like cleaning, service robotics, surveillance, warehouse stockings, etc., to numerous outdoor applications like spatial/underwater exploration, military, demining, etc. According to [4] mobile robots can be classified based on their ground-contact-based modes of locomotion as below:

- Wheeled robots
- Tracked robots
- Legged Robots
- Aerial Robots
- Snake robots
- Wheel-legged robots

Each of them have different pros and cons that make them suitable for different functions. In addition to the above mentioned mobile robots, rolling robots is a promising field that have been extensively researched in the last two decades.

Rolling robots, generally of spherical geometry, can be designed to operate holonomically. Furthermore, the spherical rolling robots cannot be overturned, while the traditional wheeled robot are ineffective when they land upside-down. A salient feature of the spherical rolling robots is that all the sensors, electronics etc can be completely sealed inside the robot which makes them ideal for hazardous environments. There are also additional functional benefits like higher strength to weight ratio, better maneuverability, and capability of operation in a wide range of terrains [5].

In the first section, we present a study on the various types of rolling robots in existence by classifying them on the basis of their propulsion mechanisms. Second section is dedicated

to the study of strategies to improve its energy efficiency. We discuss the various methods used to this end, focusing mainly on the application of *Variable Stiffness Springs (VSS)*.

2.1 Classification of Rolling Robots

For the rolling robot to move, the internal drive mechanism should be able to transfer power to the outer shell in some manner. Developing a mechanism that can provide an omnidirectional torque to the sphere (to cause any arbitrary rotation), irrespective of the orientation of either the sphere or the drive mechanism, is a difficult engineering challenge. There are multiple ways, each with its set of advantages and disadvantages, to tackle this challenge.

The absence of a single dominating drive mechanism has led to a wide variety of rolling robot designs. In this section, we discuss about the state of the art spherical robot designs by broadly classifying them into 4 categories [6]:

- Wheel based
- Barycentre offset based
- Shell deformation based
- Conservation of angular momentum based

2.1.1 Wheel Based

Initial prototypes of spherical rolling robots were wheel based designs like the *Ball Robot* developed by Halme et al. in 1996 [7]. Its internal drive unit (IDU) consists of a motor driven wheel that can be turned to steer the robot. The robot's wheel is in constant with the outer shell with the help of the spring loaded mechanism as shown in the figure 2.1. The wheel at the bottom of the sphere is motor driven and used to propel the robot. On top of the spring at the opposite end of the driven wheel is a balance wheel. The robot heading can be changed by steering the driven wheel.

Alves et al. [8] used a similar design (figure 2.2), but instead of a single wheel drive, a different four wheel drive was utilized. Generation of different motion curves was possible as the four wheels were controlled separately. Although the design was suitable for straight line motion of the spherical robot, slipping of the internal unit inside the spherical shell occurs when the robot navigates a turn which had to be managed by using a closed-loop control based in inertial sensing.

In 1997, A. Bicchi et al. proposed *Sphericle* (figure 2.3) that consists of a wheeled device enclosed in a spherical cavity to actuate motion [9] [10]. The operation of the robot for very slow speeds followed a quasi-static kinematic model that is a combination of unicycle

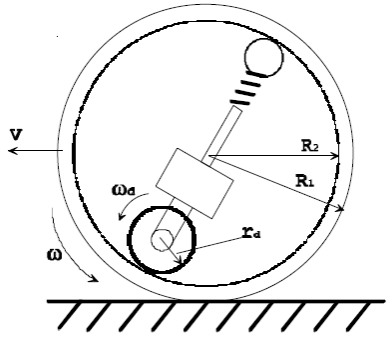
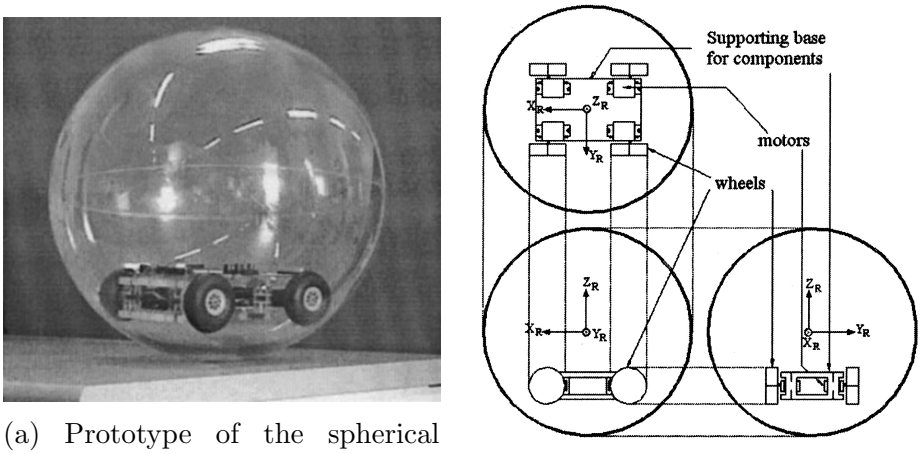


Figure 2.1: Ball Robot [7]



(a) Prototype of the spherical robot

(b) Structure of the internal unit

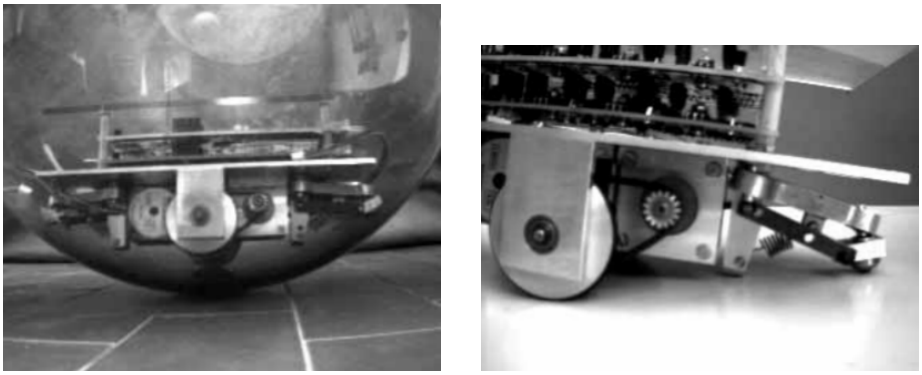
Figure 2.2: Spherical Robot Alves et al. [8]

kinematics and plate-ball kinematics. The disadvantage of the model was that the kinematics was only valid in a narrow range of operating conditions.

2.1.2 Barycentre Offset Based

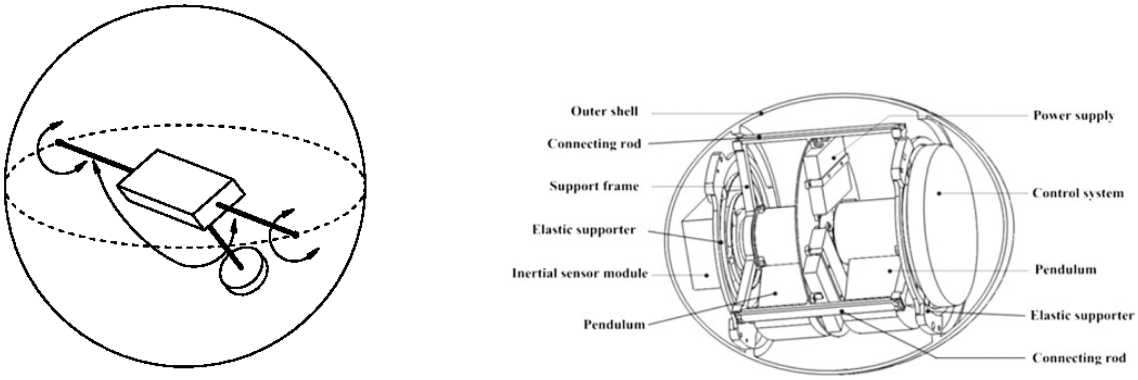
This category of spherical robots generate a desired motion by shifting their centre of mass or *barycentre*. When the barycentre is shifted from the vertical neutral axis, the robot rolls towards the new barycentre to stabilize itself. The robot can smoothly traverse its environment with proper timing and control methodologies. The main limitation to this design is its constraint on the maximum output torque.

The *pendulum driven* design is widely used in academia and industry. It was introduced by Francois et al. [11] and was commercialized by Rotundus. As shown in Figure. 2.4a, the drive mechanism consists of a pendulum located in the centre of a horizontal shaft that



(a) Laboratory prototype (b) Inner Vehicle

Figure 2.3: *Sphericle* of A. Bicchi [9]



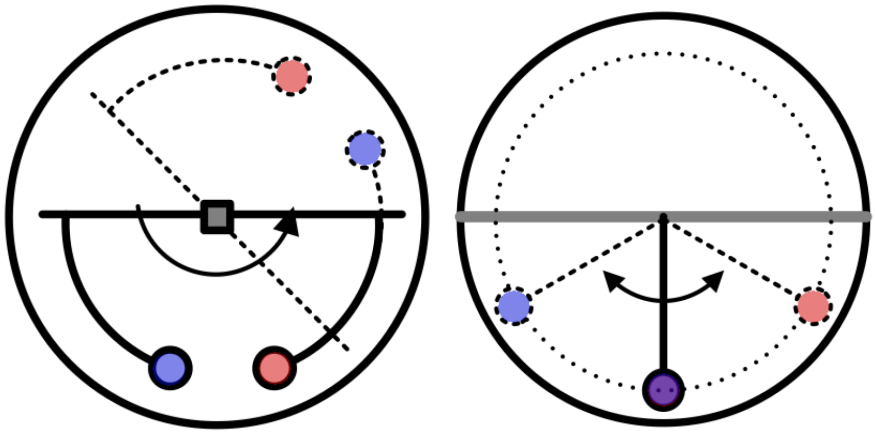
(a) Single pendulum [11] (b) Double pendulum [12]

Figure 2.4: Pendulum driven rolling robots

connects it to the spherical shell. The robot can be propelled by controlling the position of the pendulum (and thus its barycentre).

Its low-power easy to implement design outweighs its drawbacks. The commercial version of this design, Rotundus, can roll at speeds of 10 kmph, through a wide range of terrains as well as carry a payload of 1.8 kg. This design is non-holonomic, and there is a turning radius associated with its movement.

The *double pendulum* (see Figure. 2.4b) type is similar to one-pendulum type. It was proposed by Zhao et al. [12] in the design of rolling robot with an elliptical shell. The double pendulum design allows the robot to turn in place, which was not possible with simple pendulum. They demonstrated the feasibility of this design by developing a physical prototype. The robot can turn in place by following a "stick-slip" maneuver. In the stick stage, the pendulums move slowly to a horizontal position away from each other. Then during the slip stage, the pendulums are quickly forced down to their initial vertical position. This quick jerk rotates the robot on spot.



(a) Front view (b) Side view

Figure 2.5: Schematic of Kisbot II [13]

Another implementation of double pendulum is seen in Kisbot II [13] (see Figure. 2.5). The drive mechanism is designed such that the pendulums and the horizontal frame connecting them to the shell can rotate 360 degrees inside the sphere without interfering with each other. This architecture obviates the need for the stick-slip maneuver.

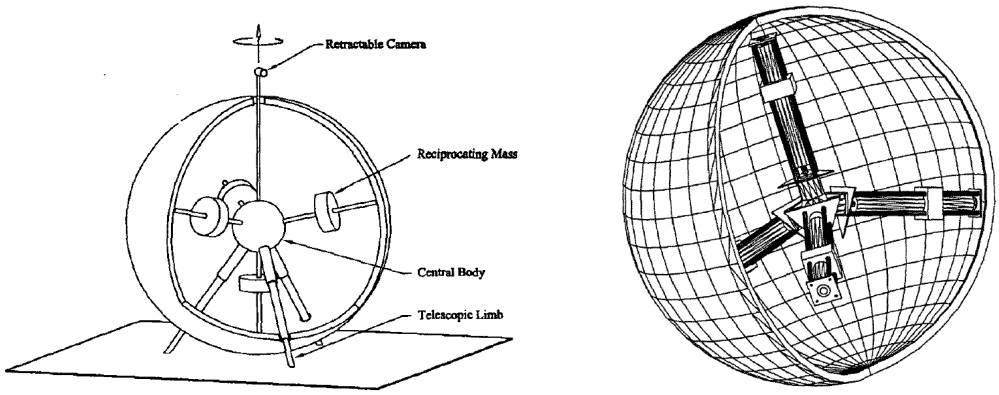
The *reciprocating mass driven* type rolling robots have a low torque requirement as the majority of its mass is located in its core, which is connected to the spherical shell using multiple shafts. The drive mechanism consists of weights, designed to traverse about these shafts, thereby shifting the centre of mass of the robot and causing the robot to roll. This design was applied in robots like *Spherobot* [5] and *August* [14].

Robots of this type are holonomic. They can move in any direction, irrespective of the orientation of the robot (within the torque limits). However, they usually operate in a low rate as the internal masses move slowly. The control of this type of robot is also challenging as one has to keep in track of the orientation as well as distance of all masses from the centre at all times.

2.1.3 Shell Deformation Based

A novel method of motion of spherical rolling robots is shell deformation. The propulsion of the robot is achieved by deformation of the outer spherical shell.

Y. Sugiyama et al. proposed a deformable robot that is capable of crawling and jumping using soft actuators [15]. Robot motion is achieved by self deformation of the shell that generates a moment by a gravitational force around the area the robot is in contact with the ground. The unstable shape in Figure 2.7b, generates a moment that causes the robot to rotate clockwise and move right. Jumping is achieved by deformation of the robot into



(a) Spherobot [5] (b) August [14]

Figure 2.6: Reciprocating mass driven rolling robots

an unstable state with high potential energy (Figure 2.7c) and its rapid release.

A type of soft actuators called *Soft Memory Alloy* (*SMA*) wires were used in the design. The prototype consisted of eight SMA coils attached to the inside of a circular rubber shell. The coil contracts when a voltage is applied to it, that in-turn deforms the outer circular rubber shell.

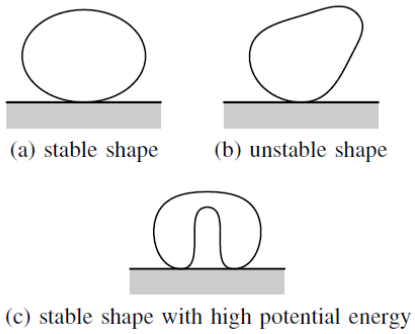
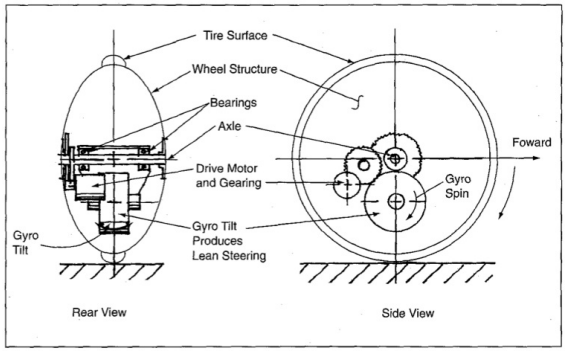


Figure 2.7: Principle of crawling and jumping (Y. Sugiyama) [15]

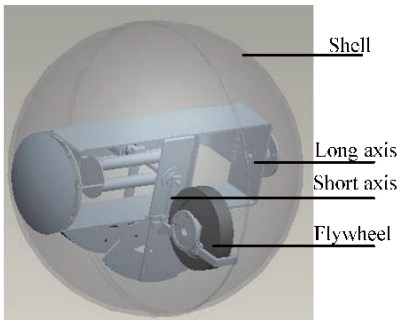
2.1.4 Conservation of Angular Momentum Based

Another novel concept that aims to overcome the drawback of limited torque output of Barycentre based designs is the Conservation of Angular Momentum (*COAM*) based spherical rolling robots. These designs employ the concept of *Control Moment Gyroscope* (*CMG*) that exploits the relation between the output torque and the angular velocity of the CMG.

The first COAM based design was *Gyrover*, a single-wheel robot that exploits gyroscopic forces for steering and stability developed by B. Brown et al. in 1997 [16]. The steering of



(a) *Gyrover* Configuration [16]



(b) Model of `BYQ-V` [17]

Figure 2.8: COAM based robots

the *Gyrover* was based on the principle of gyroscopic precession, a spinning wheel tends to precess at right angles to an applied torque because of its angular momentum. An additional internal gyroscope was used for stability.

Qinqxuan et al. developed a CMG based robot that used a pendulum type drive system to supplement the robot stability [17]. The design incorporated the pendulum drive based design with the pendulum bob implementing CMG concept (Figure 2.8b). The rotation of the robot is achieved by CMG rotation, while the output torque of the robot aided by the effects of precession torque of a CMG is greater than the torque that a normal pendulum drive could generate.

The different classes of Spherical rolling robots have their distinct advantages and drawbacks. Wheel based rolling robots can be designed to be holonomic or non-holonomic depending on the wheel configuration. The barycentre based designs are popular due to their simple but effective model. However, their limited torque output is a major hindrance for real world applications. The Shell deformation based designs is a promising field that can tackle rougher terrains but they are limited by the small force that the soft actuators can generate. Designs based on COAM is another active field of research but is restricted by the lack of continuous torque output.

2.2 Variable Stiffness Actuators

Energy efficiency is an integral factor to be considered while designing a mobile robot. There are several strategies to improve the energy efficiency. The first is to minimize the mass and inertia of the robot by carefully designing the robot. Secondly, a considerable amount of energy can be conserved by optimizing the path planning. The shortest path is not always the most energy efficient. Finally, we can try to optimize the actuators to reduce energy.

The conventional actuators are inexpensive, efficient and easy to control in normal situations.

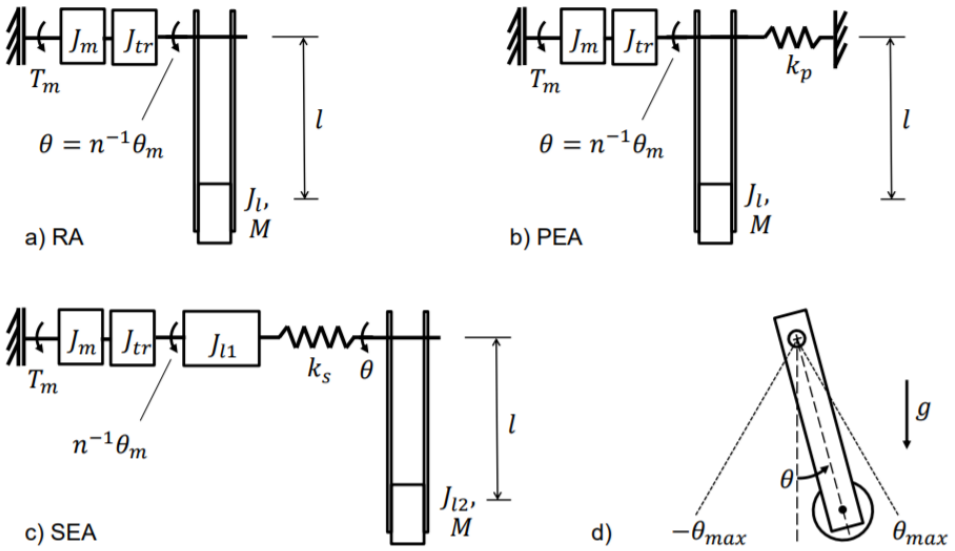


Figure 2.9: Considered actuator topologies: (a) Rigid Actuator, (b) Parallel Elastic Actuator, (c) Series Elastic Actuator. Front view of the pendulum is shown in (d). [20]

However, they are not good interacting with the environment, which is inherent to any kind of locomotion. An actuator which has a compliant element in it will be ideal for this application. This interest has lead to the growing effort in the development of variable stiffness actuators in recent years.

The first VSA prototype for safe and fast motion was developed by Bicci et al. [18]. Since then, several groups have designed VSAs, with elastic elements storing energy in addition to altering stiffness. VSAs can be classified [19] into three major groups:

- **Spring preload**: The stiffness of this category of VSAs is controlled by altering the spring preload. This could be done with antagonistic springs with antagonistic or independent motors or with just a simple spring with a motor to control its stiffness.
- **Transmission ratio**: In this category, the transmission ratio between the load and the elastic elements is altered to control the stiffness of the actuator. The advantage of this type is that, the system at equilibrium requires no power to keep a constant stiffness. It is usually implemented by controlling the configuration of the lever mechanism, the properties of a non-linear mechanical link or the transmission ratio of a continuously variable transmission.
- **Physical properties**: This category of VSAs modifies the effective physical structure of the spring to achieve variations in stiffness. It is usually achieved by varying the cross-sectional area or the active spring length of the actuator.

Another major design consideration is the location of the compliant component with respect to the rest of the system, i.e. the actuator, the ground and the load. The compliant element

can be connected in series or parallel to the actuating element and the resulting designs are called *Series Elastic Actuators (SEA)* and *Parallel Elastic Actuators (PEA)* respectively.

Verstraten et al. [20] performed a detail analysis of the power and energy consumption of these actuators by imposing a sinusoidal motion to a pendulum load. They demonstrated that both SEA and PEA led to a decrease in peak power and peak energy consumption, given the stiffness of the elastic element is tuned properly. Their experiments confirmed an energy reduction upto 78% (SEA) and 20% (PEA) compared to rigid actuators.

The SEA arrangement can affect the accuracy for high speed robot applications as they lead to uncontrolled robot deflections [1]. Therefore, we decided to use Variable Stiffness Springs (VSS) in parallel configuration to actuate our system. The benefits of this choice can be summarized as: i) reduction in input motor torques by controlled release of the stored potential energy and ii) direct power connection between the motor and robot links leads to increased accuracy.

As we have seen, rolling robots is an active field of research with immense potential. However, energy efficiency is a major barrier that hinders advancements in this field. Our aim in this project is to improve the energy efficiency of the robot by using VSS in parallel configuration. In the following chapter we derive the analytical models of three reciprocating mass based rolling robots to study the effect of parallel elastic actuation in their energy consumption.

Chapter 3

Analytical Modelling

In this chapter, we perform a detail study on a reciprocating mass based rolling robot. We model three types of robots: (i) a nominal robot with no elastic elements, (ii) a robot with fixed stiffness springs in parallel and, (iii) a robot with VSS in parallel.

As explained in Section. 2.1.2, the robot consists of a spherical shell, a tetrahedral frame centred at the centre of the shell, and masses that can move along each link of the frame (See Fig. 3.1). The position of the *barycentre* of the robot can be controlled by moving the reciprocating masses. This causes an imbalanced moment that propells the robot.

3.1 Rolling Robot without Elastic Elements (Nominal)

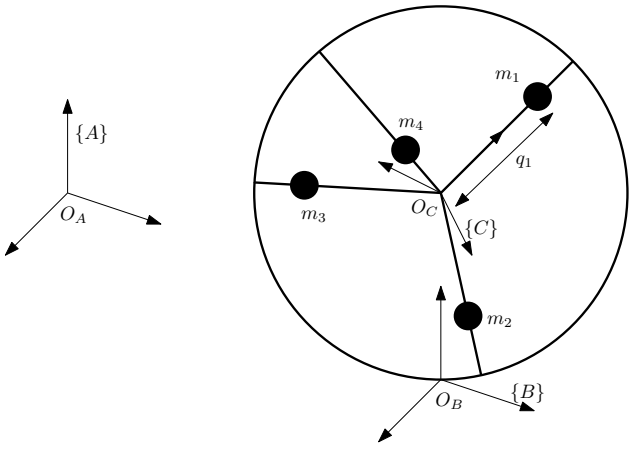


Figure 3.1: Schematic of the rolling robot without elastic elements: In this type of actuation, q_i can be controlled to propel the robot.

We define three coordinate systems as shown in Fig. 3.1. $\{A\}$ is a global coordinate system

with its origin at O_A, and z-axis along the vertical. The coordinate system $\{B\}$ has its origin, O_B, located at the point of contact of the shell with the ground. Its axes are aligned along those of $\{A\}$. The origin of $\{C\}$ is located at the centre of the sphere. Its z-axis is aligned with the axis of reciprocation of Mass 1. The axis of motion of Mass 2 lies on the x-z plane of $\{C\}$.

The kinematic and dynamic equations of the systems were developed in `Mathematica` (symbolic computation software). The derivation of those equations are explained in detail in the following sections.

3.1.1 Geometric Model

The robot is assumed to have a pure rolling motion with respect to the ground. The point of contact with the ground is always at rest. This makes $\{B\}$ an inertial coordinate system, and thus we perform our dynamic analysis with respect to this system.

We need 7 variables to fully represent the robot in $\{B\}$ (4 for the displacement of masses from the and 3 for the orientation of the $\{C\}$ w.r.t $\{B\}$). The state vector of the robot:

$$\mathbf{q} = [q_1, q_2, q_3, q_4, \alpha, \beta, \gamma]^\top \tag{3.1}$$

where, q_i $(i = 1, \ldots, 4)$ is the displacement of i^{th} mass from the centre, α, β, γ are the roll-pitch-yaw angles. The position of the centre of the shell:

$${}^B\mathbf{p}_{O_c} = [0, 0, R]^\top \tag{3.2}$$

where R is the radius of the sphercial shell.

The unit vectors along the tetrahedral spokes are defined as:

$$\begin{aligned} {}^C\hat{\mathbf{p}}_1 &= [0, 0, 1]^\top \\ {}^C\hat{\mathbf{p}}_2 &= \mathbf{R}_Y(\phi)\ {}^C\hat{\mathbf{p}}_1 \\ {}^C\hat{\mathbf{p}}_3 &= \mathbf{R}_Z(2\pi/3)\ {}^C\hat{\mathbf{p}}_2 \\ {}^C\hat{\mathbf{p}}_4 &= \mathbf{R}_Z(2\pi/3)\ {}^C\hat{\mathbf{p}}_3 \end{aligned} \tag{3.3}$$

where, $\mathbf{R}_Y$ and $\mathbf{R}_Z$ represent rotations about Y and Z axes respectively and ϕ ($\approx 109.47°$)is the *tetrahedral angle*.

The position vectors of the masses in $\{C\}$ are defined as:

$${}^C\mathbf{p}_i = q_i\ {}^C\hat{\mathbf{p}}_i; \qquad i = 1, \ldots, 4 \tag{3.4}$$

where, q_i is the displacement of mass i from the centre of the sphere.

The positions of these masses in $\{B\}$ are defined as:

$$\begin{aligned} {}^{B}\mathbf{p}_i &= {}^{B}\,\mathbf{p}_{O_c} + {}^{B}\,\mathbf{R}_C(\alpha,\beta,\gamma)\ {}^{C}\mathbf{p}_i \\ &= {}^{B}\,\mathbf{p}_{O_c} + q_i\,{}^{B}\hat{\mathbf{p}}_i; \qquad i = 1,\ldots,4 \end{aligned} \tag{3.5}$$

where, ${}^{B}\mathbf{R}_C(\alpha,\beta,\gamma)$ is the rotation matrix using Z-Y-X Euler angle convention, and ${}^{B}\hat{\mathbf{p}}_i$ is ${}^{C}\hat{\mathbf{p}}_i$ represented in $\{B\}$. ${}^{B}\mathbf{p}_i$ was found to be a function of q_i, α, β and γ.

3.1.2 First Order Kinematic Model

The angular velocity of the spherical shell about $\{B\}$,

$$\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\top \tag{3.6}$$

The velocity of the masses in $\{B\}$:

$$\begin{aligned} {}^{B}\mathbf{v}_i &= \boldsymbol{\omega} \times^{B} \mathbf{r} + \mathbf{v}_{rel} \\ &= \boldsymbol{\omega} \times^{B} \mathbf{r}_i + \dot{q}_i\ {}^{B}\hat{\mathbf{p}}_i \qquad i = 1,\ldots,4 \end{aligned} \tag{3.7}$$

where, ${}^{B}\mathbf{r}_i$ is the position vector of Mass i, $\mathbf{v}_{rel}$ is ${}^{C}\mathbf{v}_i$ expressed in $\{B\}$, $\dot{q}_i$ is the translational velocity of Mass i along its axis. ${}^{B}\mathbf{v}_i$ is a function of $q_i, \dot{q}_i, \alpha, \beta, \gamma$, and $\boldsymbol{\omega}$.

3.1.3 Second Order Kinematic Model

Similarly, the angular accelerations of the spherical shell about $\{B\}$,

$$\dot{\boldsymbol{\omega}} = [\dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z]^\top \tag{3.8}$$

The accelerations of the masses:

$$\begin{aligned} {}^{B}\mathbf{a}_i &= \dot{\boldsymbol{\omega}} \times^{B} \mathbf{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times q_i\,{}^{B}\hat{\mathbf{p}}_i) + 2(\boldsymbol{\omega} \times \mathbf{v}_{rel}) + \mathbf{a}_{rel} \\ &= \dot{\boldsymbol{\omega}} \times^{B} \mathbf{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times q_i\,{}^{B}\hat{\mathbf{p}}_i) + 2(\boldsymbol{\omega} \times \dot{q}_i\ {}^{B}\hat{\mathbf{p}}_i) + \ddot{q}_i\ {}^{B}\hat{\mathbf{p}}_i \qquad i = 1,\ldots,4 \end{aligned} \tag{3.9}$$

where, $\mathbf{a}_{rel}$ is ${}^{C}\mathbf{a}_i$ expressed in $\{B\}$, $\ddot{q}_i$ is the acceleration of Mass i along its axis. ${}^{B}\mathbf{a}_i$ is a function of $q_i, \dot{q}_i, \ddot{q}_i, \alpha, \beta, \gamma, \boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$.

For the static case, the accelerations of the masses can be approximated as:

$${}^{B}\mathbf{a}_i \approx \ddot{q}_i\ {}^{B}\hat{\mathbf{p}}_i \tag{3.10}$$

This approximation is reasonable as we operate in small velocities.

3.1.4 Dynamic Model

In this section, we study the the effect of different different forces, namely the actuator forces and gravity on the robot. For this, we first define the various mass parameters of the robot. The reciprocating bodies are assumed to be point masses with mass m and zero moment of inertia. The spherical shell has a mass M and radius R.

Its moment of inertia about its centre (the thickness of the shell is assumed to be negligible compared to its radius) is $\frac{2}{3}MR^2$. Using parallel axis theorem, we find the moment of inertia of the shell about the origin of $\{B\}$ as:

$$I_{shell} = \begin{bmatrix} \frac{5}{3}MR^2 & 0 & 0 \\ 0 & \frac{5}{3}MR^2 & 0 \\ 0 & 0 & \frac{2}{3}MR^2 \end{bmatrix} \tag{3.11}$$

At any instant, the inertial and gravitational forces acting on the masses, along the direction of the spoke, are balanced by the actuators. Therefore, the force applied by the actuators can be found as:

$$\begin{aligned} F_i &= {}^B\,\hat{\mathbf{q}}_i.(\mathbf{F}_{inertia,i} + \mathbf{F}_{gravity,i}) \\ &= m\,{}^B\hat{\mathbf{q}}_i.(-\mathbf{g} + {}^B\,\mathbf{a}_i) \end{aligned} \tag{3.12}$$

where, $\mathbf{g} = [0, 0, -g]^\top$ is the gravity vector.

The force balance about O_B would give us the reaction forces at B. We omit that computation since it is not relevant for our application. We now compute the moment balance about O_B. The total gravity moment:

$$\mathbf{M}_{gravity} = \sum_{i=1}^{4} \mathbf{M}_{gravity,i} = \sum_{i=1}^{4} m({}^B\mathbf{p}_i \times \mathbf{g}) \tag{3.13}$$

Similarly, the inertia moment due to the masses,

$$\mathbf{M}_{inertia,masses} = -m \sum_{i=1}^{4} {}^B\mathbf{p}_i \times \mathbf{a}_i \tag{3.14}$$

The inertia moment due to the shell,

$$\mathbf{M}_{inertia,shell} = \mathbf{I}_{shell}\ \dot{\boldsymbol{\omega}} + (\boldsymbol{\omega} \times (\mathbf{I}_{shell}\ \boldsymbol{\omega})) \tag{3.15}$$

The total inertia moment:

$$\mathbf{M}_{inertia} = \mathbf{M}_{inertia,shell} + \mathbf{M}_{inertia,masses} \tag{3.16}$$

Inertia balance about O_B gives us:

$$\mathbf{M}_{inertia} + \mathbf{M}_{gravity} = 0 \tag{3.17}$$

Equation 3.17 gives us the relationship between the rolling motion of the robot and the translation motion of the reciprocating masses.

For the static case, this equation can be approximated as:

$$I_{shell}\dot{\boldsymbol{\omega}} = -\sum_{i=1}^{4} m({}^{B}\mathbf{p}_i \times \mathbf{g}) + m\sum_{i=1}^{4} {}^{B}\mathbf{p}_i \times (\ddot{q}_i \, {}^{B}\hat{\mathbf{p}}_i) \tag{3.18}$$

3.2 Rolling Robot with Fixed Stiffness Springs in Parallel

In this section, we discuss about the modelling of a spherical rolling robot with parallel compliant actuation. The schematic of this robot is shown in Fig. 3.2. The masses are now connected through a spring with spring constant, k, free length, l_s and initial contact point at a distance of q_s from the centre.

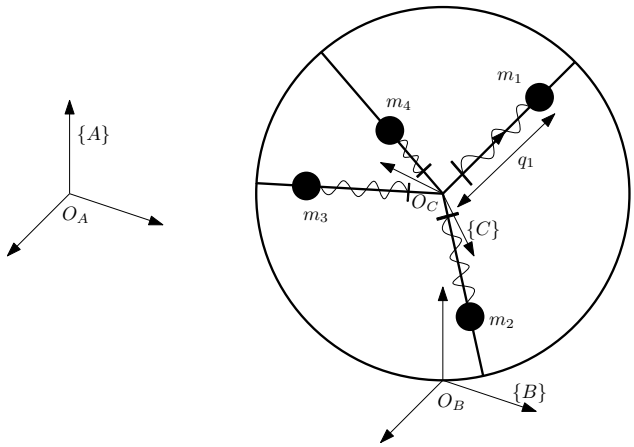


Figure 3.2: Schematic of the rolling robot with compliant actuation: As before, q_i can be controlled to propel the robot. The fixed stiffness spring, located at q_s, stores and releases energy during motion.

This robot is very similar to the one discussed in Section. 3.1. Since the springs just induce an internal force to the system, it doesn't affect the dynamics of the robot. Therefore, the dynamics of this robot is identical to that of the one in Section. 3.1. However, the actuator forces change as we now have a spring in parallel. The actuator force of i^{th} actuator:

$$\begin{aligned} F_i &= \hat{\mathbf{q}}_i.(\mathbf{F}_{inertia,i} + \mathbf{F}_{gravity,i}) + \mathbf{F}_{spring,i} \\ &= m\hat{\mathbf{q}}_i.(-\mathbf{g} + {}^{B}\,\mathbf{a}_i) + k(q_i - q_s - l_s) \end{aligned} \tag{3.19}$$

where, q_i is the displacement of the mass from the centre and in this case q_s is a constant. One may note that the state vector of this system has the same dimension as of that of the robot without spring.

3.3 Rolling Robot with VSS in Parallel

In this section, we add a variable stiffness spring to the system. The schematic is shown in Fig. 3.3. The new system is very similar to that defined in Section. 3.2, the only difference being that we use a Variable Stiffness Spring instead in this case. The stiffness of the spring is varied by altering the initial point of the spring q_{s_i}. In future, we hope to enhance the performance by exploiting this property of this robot.

The state space of the robot has 11 dimensions and the state vector $\mathbf{q}$ is represented as:

$$\mathbf{q} = [q_1, q_2, q_3, q_4, q_{s_1}, q_{s_2}, q_{s_3}, q_{s_4}, \alpha, \beta, \gamma]^\top \tag{3.20}$$

where, q_i and q_{s_i} are the position of the masses and spring bases of i^{th} link respectively and α, β, and γ the roll, pitch and yaw angles.

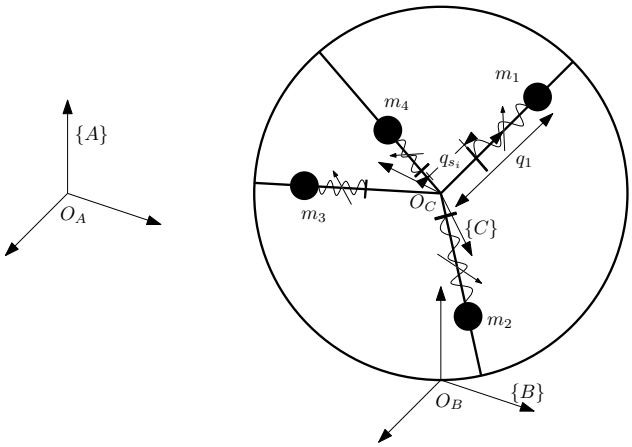


Figure 3.3: Schematic of the rolling robot with VSS: In this case, we can control the position of the mass, q_i, and spring base, q_{s_i}, located in the i^{th} link.

Since the mass of the spring base (m_s) is assumed to be negligible compared to the rest of the system, the dynamics of this robot is identical to that of the robot in Section. 3.1. The actuator force at i^{th} actuator:

$$\begin{aligned} F_i &= \hat{\mathbf{q}}_i.(\mathbf{F}_{inertia,i} + \mathbf{F}_{gravity,i})\mathbf{F}_{spring,i} \\ &= m\hat{\mathbf{q}}_i.(-\mathbf{g} +^B \mathbf{a}_i) + k(q_i - q_{s_i} - l_s) \end{aligned} \tag{3.21}$$

The dynamics of the spring system is given by the relation:

$$F_{s_i} = m_s\ddot{q}_i + h_s(q_s, \dot{q}_s) + k(q_{s_i} + l_s - q_i) \tag{3.22}$$

Chapter 4

Validation

Since the equations of motion of the model are lengthy and complicated, there is a good chance of encountering errors in modelling. Therefore, it is of utmost importance to validate the dynamic model. In this chapter, we talk about the validation of the dynamic models derived in Chapter 3.

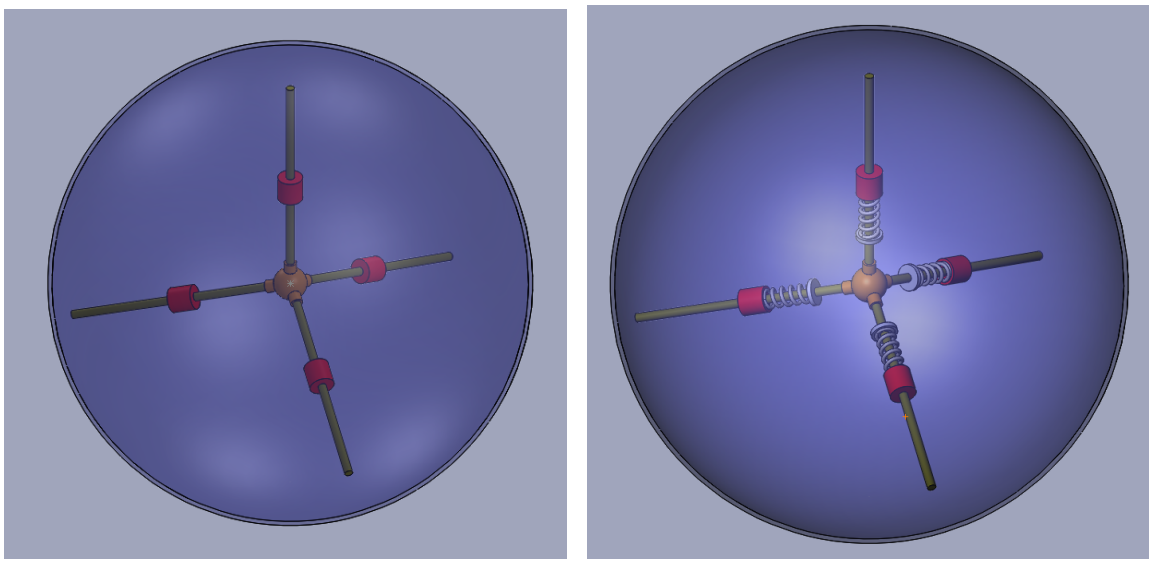
We used the software, `MSC ADAMS`, which is the benchmark for dynamic simulations. In the following sections, we detail about the development of the ADAMS models and validation experiments.

4.1 SolidWorks model

Since the CAD modelling capability of ADAMS is limited, we first developed the CAD model of the robot in `SolidWorks`. The following are the components used for the model:

- Spherical shell
- Tetrahedral frame
- Reciprocating masses
- Spring base

The nominal rolling robot was modelled as shown in Figure. 4.1a. The robots with parallel compliant elements have provisions to place the springs. The CAD model of those robots are shown in Figure. 4.1b. The outer radius of the spherical shell, R, was chosen as 290 mm.



(a) Rolling robot without elastic elements (b) Rolling robot with spring/VSS

Figure 4.1: CAD models of the rolling robots

4.2 ADAMS Model

CAD models developed in `SolidWorks` were then imported as a *parasolid* file (.x_t) to ADAMS. We used *parasolid* format as it models geometric features with exact curvature. This model preserves the curvature of the spherical shell as it doesn't convert it to an edged surface. This leads to an accurate modelling of the system, the simulations were observed to be highly noisy when other formats were chosen. The downside is that the simulation time is relatively high when using *parasolid* model.

As usual, the high accuracy comes with a price. The duration of simulations increases significantly with this choice. However, this is a fair trade-off. The solver settings have to be changed to interpret exact features for the contact and simulation. This was done by selecting the option: Settings>>Solver>>Contacts>>Parasolid.

This format is convenient because one file contains all the geometry and `ADAMS/Exchange` creates a separate part for each solid. In the next section we detail about the different parameters to be set in order to employ the ADAMS model for validation/simulation.

4.2.1 Model Properties and Parameters

On importing the CAD model, ADAMS creates separate bodies for the four masses, four spokes, core, sphere, base and spring base (in the case of model with VSS). We update the mass properties of the bodies as given in Table. 4.1. The other relevant parameters

are tabulated in Table. 4.2. The two ADAMS models are shown in Figure 4.2. Additional information on the ADAMS model description is provided in the Appendix 1.

Table 4.1: Mass properties: `ADAMS` model

Body	**Mass**(kg)	**Inertia**$(kg.mm^2)$
Masses	1	1.0×10^{-09}
Spokes	0	-
Core	0	-
Sphere	5	2.8033×10^{05}
Spring Base	0	-

Table 4.2: Relevant parameters: `ADAMS` model

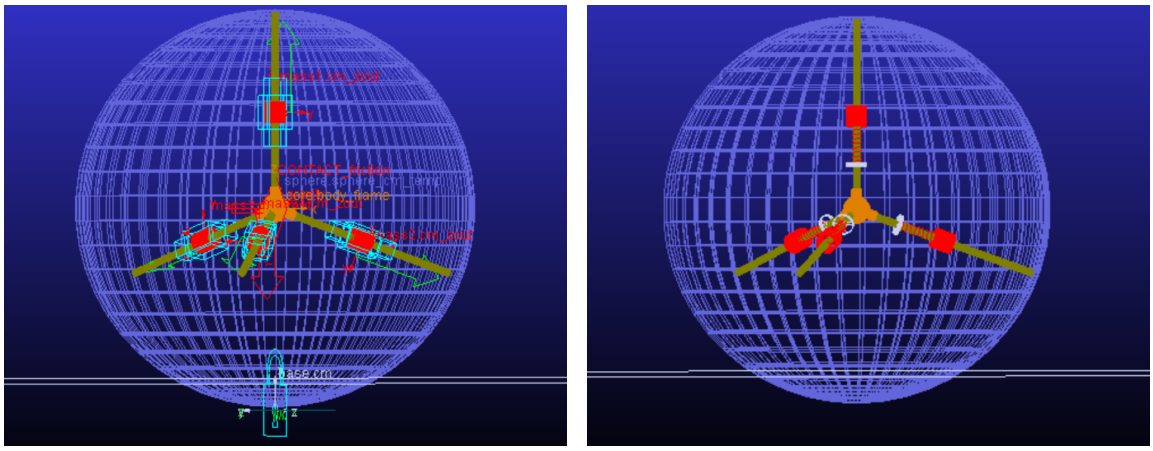
Parameters	**Values**
Initial position of masses	140 mm
Initial position of spring base	70 mm
Free length of the spring	70 mm
Spring stiffness	$7.5 \times 10^{-03} N.mm^{-1}$
Damping	0
Preload length	0

To get a satisfactory representation of the ADAMS model of the rolling robot it is important to set the contact forces between the sphere and the base accurately. Table 4.3 furnishes information on the contact model configuration set in ADAMS. Documentation relating to the contact model can be found in Appendix 1.1.

4.2.2 Simulation and Solver Settings

The choice of proper solver and step size is important to obtain a simulation with reliable results. We can access the ADAMS Solver settings under the Simulation Control window. Instead of the default ADAMS settings that set the simulation to 50 *Steps* independent of the simulation time, choosing *Step Size* of 0.01 will ensure a better control on the simulation time and satisfactory accuracy.

The Solver Settings - *Dynamic*, by default is set to `GSTIFF` *Integrator* and `I3` *Formulation* with an error of 10^{-3}. This configuration was found to give competent results. Though the use of `SI2` *Formulation* was observed to reduce the simulation time, it failed occasionally when attempting `ADAMS/Simulink` co-simulation. A critical settings to be modified is that of *Contact* settings of the solver. The *Geometric Library* must be changed from default to *Parasolids*, to obtain meaningful simulation results.



(a) Rolling robot without elastic elements (b) Rolling robot with spring/VSS

Figure 4.2: ADAMS models of the rolling robots

Table 4.3: ADAMS Contact Model Parameters

Parameter	Option set
Normal Force	Impact
Stiffness	1.0×10^{08}
Force Exponent	2.2
Damping	1.0×10^{04}
Penetration Depth	1.0×10^{-04}
Frictional Force	Coulomb
Static Coefficient	0.3
Dynamic Coefficient	0.1
Stiction Trans. Vel	0.1
Friction Trans. Vel	1.0

We use the ADAMS-plugin called ADAMS/Controls to export the ADAMS-subsytem that uses the state variables (system elements) to interact with the Matlab/SIMULINK.

4.3 Experiments

A set of experiments were devised to validate the dynamic model. The robots were made to traverse a trajectory and the following data was recorded:

1. Displacement of the moving masses from the centre and their derivatives $(q_i, \dot{q}_i, \ddot{q}_i)$
2. Euler angles of $\{C\}$ w.r.t $\{A\}$ (α, β, γ)
3. Angular velocity and acceleration of the robot $(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}})$

4. Cartesian positions, velocities and accelerations of the reciprocating masses $(\mathbf{p}_i, \mathbf{v}_i, \mathbf{a}_i)$
5. Forces acting on the reciprocating masses (F_i)
6. Forces acting on spring bases (F_{s_i}) (if applicable)

The Cartesian positions, velocities, accelerations, and forces recorded in `ADAMS` was compared to those from the dynamic model.

The experiments for first two types of rolling robots were defined as follows:

1. An acceleration of $40\cos(\frac{2\pi t}{T_1})$, with the time period, T_1, of 5 seconds was given to Mass 2.
2. An acceleration of $40\cos(\frac{2\pi t}{T_1})$, with the time period, T_1, of 5 seconds was given to Mass 2 and Mass 3, and an acceleration of $20\cos(\frac{2\pi t}{T_2})$, with time period of,$T_2$, of 10 seconds was given to Mass 1 and Mass 4.

The model of the rolling robot with parallel VSS was validated by performing the following experiments:

1. An acceleration of $40\cos(\frac{2\pi t}{T_1})$, with the time period, T_1, of 5 seconds was given to Mass 2. An acceleration of $-5\cos(\frac{2\pi t}{T_{1,s}})$, with time period $T_{1,s}$ of 10 seconds, was given to the spring base of link 2.
2. An acceleration of $40\cos(\frac{2\pi t}{T_1})$, with the time period, T_1, of 5 seconds was given to Mass 2 and Mass 3, and an acceleration of $20\cos(\frac{2\pi t}{T_2})$, with time period of,$T_2$, of 10 seconds was given to Mass 1 and Mass 4. An acceleration of $-5\cos(\frac{2\pi t}{T_{1,s}})$, with $T_{1,s}$ of 10 seconds given to the bases of link 1,2, and 4, and an acceleration of $5\cos(\frac{2\pi t}{T_{2,s}})$, with $T_{2,s}$ of 5 seconds, given to spring base of link 3.

4.4 Results

4.4.1 Rolling Robot without Elastic Elements (Nominal)

The model was found to perform well for the two tests. The accuracy of the model can be seen from the plots on the geometric model (see Figure. 4.3), the kinematics model (see Figure. 4.4) and the dynamic model (see Figure. 4.5). As you may observe, our model is very accurate with error of the order of 10^{-5}.

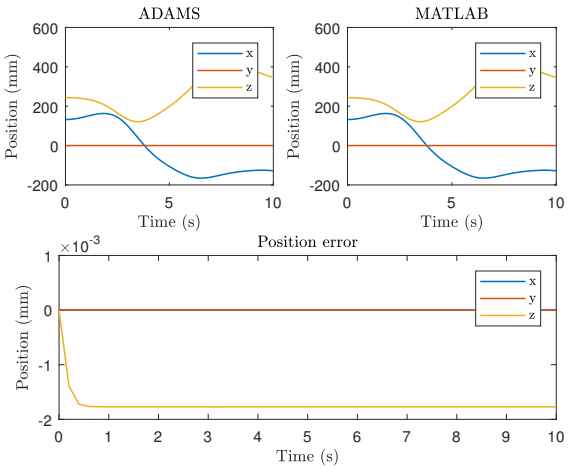
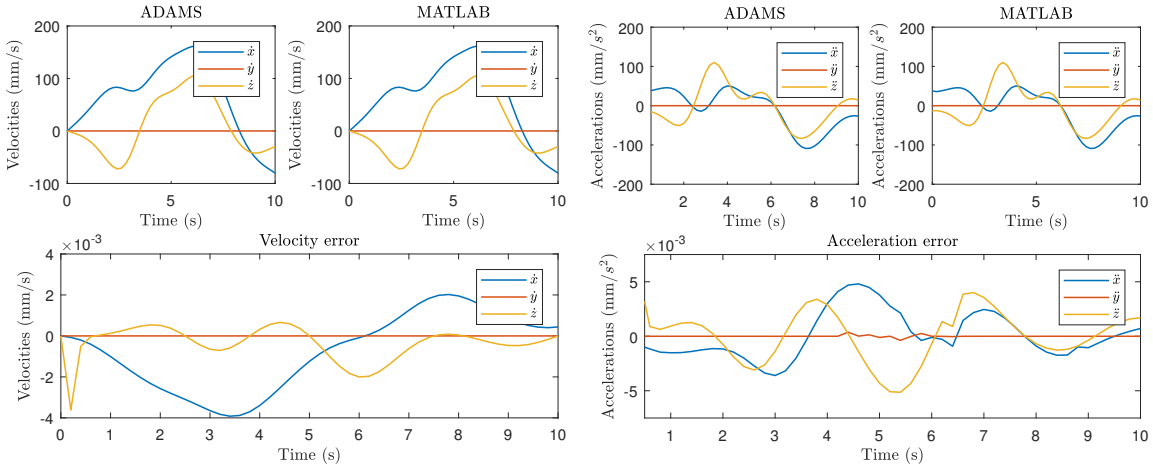


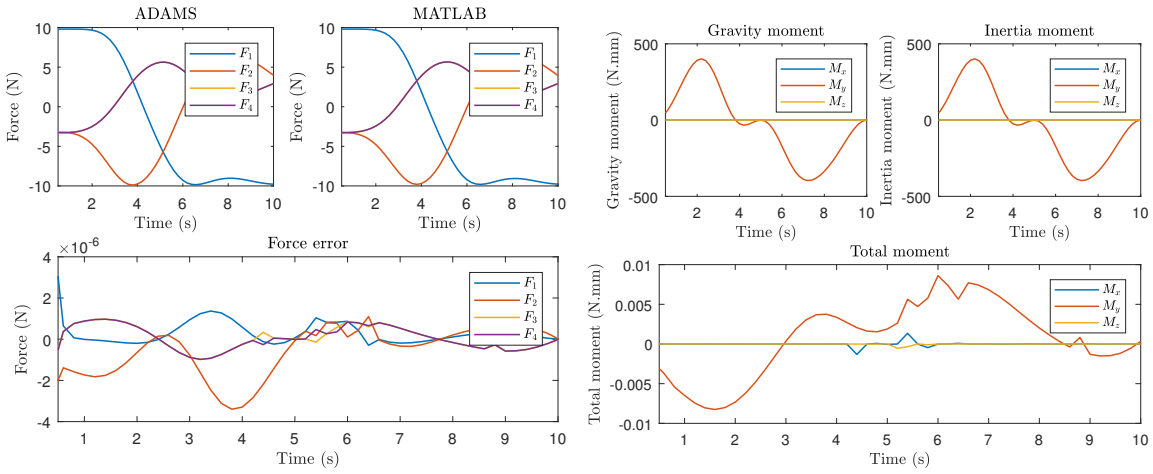
Figure 4.3: Validation of the geometric model: Here, we compare the Cartesian position of Mass 2 in $\{B\}$, ${}^{B}\mathbf{p}_2$, obtained from ADAMS with that from the geometric model while performing experiment 1.



(a) Cartesian velocity of Mass 2, ${}^{B}\mathbf{v}_2$, during experiment 1

(b) Cartesian acceleration of Mass 2, ${}^{B}\mathbf{a}_2$, during experiment 1

Figure 4.4: Validation of the kinematics model by comparing with the results from ADAMS

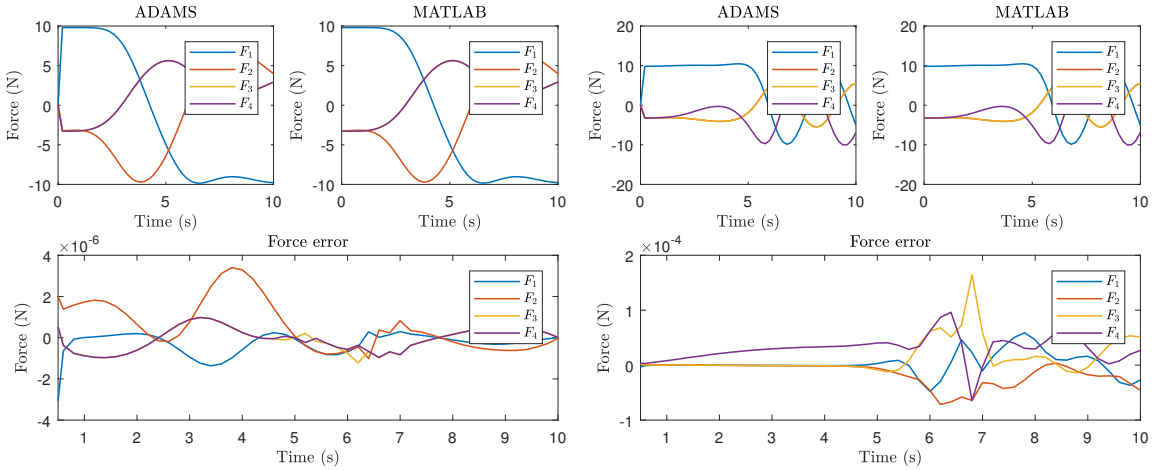


(a) Forces on masses F_i in experiment 1 (b) Moments in experiment 1

Figure 4.5: Validation of dynamic model of the nominal robot

4.4.2 Rolling Robot with Fixed Stiffness Springs in Parallel

The model was validated with a high accuracy as shown in Figure. 4.6. The error is of the order of 10^{-6}.

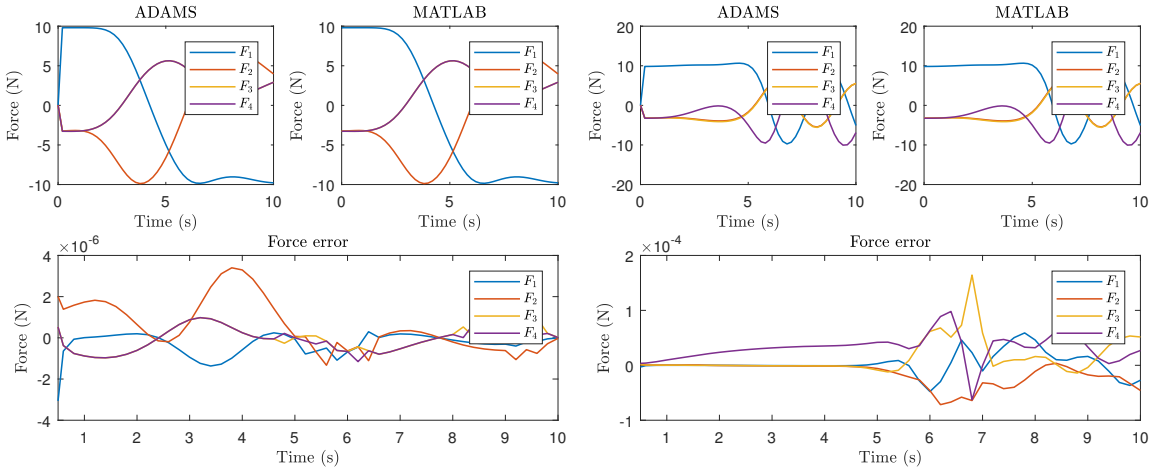


(a) Forces on masses in experiment 1 (b) Forces on masses in experiment 2

Figure 4.6: Validation of the dynamic model of the robot with constant stiffness spring in parallel

4.4.3 Rolling Robot with Parallel VSS

The results of this model is similar to that of the previous one. It was validated with an error of the order of 10^{-6}.



(a) Forces on masses in experiment 1

(b) Forces on masses in experiment 2

Figure 4.7: Validation of the dynamic model of the robot with VSS in parallel

Chapter 5

Control and Trajectory Generation

In this chapter, we address the problem of motion planning. The problem is defined as the task of moving the centre of the rolling robot from an initial point to a final point. We present an algorithm for steering the spherical robot by continuously changing the location of its centre of mass. We do not consider the dynamics of the robot as it is prohibitively expensive.

First, we introduce a simple control strategy to generate a generate moments about X and Y axes. Later, we use this moment for motion planning of the robot.

5.1 Moment Generation

5.1.1 Position Control of Masses

As mentioned before, the robot is propelled by 4 reciprocating masses. The simple PD controller to control the position of the masses can be described as:

$$\ddot{q}_i = -k_p q_i - k_d \dot{q}_i \tag{5.1}$$

The gains, k_p and k_d were chosen such that the tracking error was less than 0.5 mm.

5.1.2 Steering Algorithm

In this section, we propose an algorithm for steering the robot by continuously modifying the locations of the moving masses. The positions of the centre of mass of the robot in local

frame is described as:

$$^{B}\mathbf{p}_{COM} = \frac{1}{4}\sum_{i=1}^{4} {}^{B}\mathbf{p}_{i} \tag{5.2}$$

where, $^{B}\mathbf{p}_{i}$ is defined in Equation. 3.4.

The moment generated due to the weight of the masses about $\{B\}$ are given by the relation:

$$\begin{aligned}\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} &= 4m\,{}^{B}\mathbf{p}_{COM} \times \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \\ &= 4mg \begin{bmatrix} -{}^{B}y_{COM} \\ {}^{B}x_{COM} \\ 0 \end{bmatrix}\end{aligned} \tag{5.3}$$

where, mg is the weight of the masses and $\times$ is the cross product. You may observe that, $M_z = 0$. This means that the robot cannot generate a spin about the vertical axis.

By controlling q_i, we can control $^{B}\mathbf{p}_{COM}$ and thus generate desired M_x and M_y. Since $^{B}x_{COM}$ and $^{B}y_{COM}$ are the only terms that effect the moments, we just have to control them. However, we have the freedom to control the positions of 4 masses.

We use the extra degrees of freedom to ensure that the masses stay as close as possible to the initial position. The following 4 equations were solved to find the positions of the masses.

$$\begin{aligned} {}^{B}x_{COM} &= x_{desired} \\ {}^{B}y_{COM} &= y_{desired} \\ {}^{B}z_{COM} &= R \\ \sum_{i=1}^{4} q_i &= 4q_{init} \end{aligned} \tag{5.4}$$

where, R is the radius of the spherical shell and q_{init} is the home position of the masses. The positions of the masses were found as:

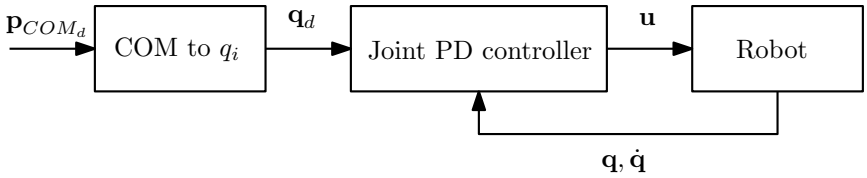


Figure 5.1: Joint control scheme

$$
\begin{aligned}
q_1 =& q_{init} + 3x\sin(\alpha)\sin(\gamma) + 3\cos(\alpha)(x\sin(\beta)\cos(\gamma) - y\sin(\gamma)) \\
&+ 3y\sin(\alpha)\sin(\beta)\cos(\gamma) \\
q_2 =& q_{init} - x\sin(\alpha)\sin(\gamma) + \cos(\alpha)\left(-x\sin(\beta)\cos(\gamma) + 2\sqrt{2}x\cos(\beta) + y\sin(\gamma)\right) \\
&- y\sin(\alpha)\sin(\beta)\cos(\gamma) + 2\sqrt{2}y\sin(\alpha)\cos(\beta) \\
q_3 =& q_{init} - x\sin(\alpha)\sin(\gamma) - \sqrt{6}x\sin(\alpha)\cos(\gamma) \\
&+ \cos(\alpha)\left(-\sqrt{2}x\cos(\beta) + \sin(\gamma)\left(\sqrt{6}x\sin(\beta) + y\right) + \cos(\gamma)\left(\sqrt{6}y - x\sin(\beta)\right)\right) \\
&+ \sqrt{6}y\sin(\alpha)\sin(\beta)\sin(\gamma) - y\sin(\alpha)\sin(\beta)\cos(\gamma) - \sqrt{2}y\sin(\alpha)\cos(\beta) \\
q_4 =& q_{init} - x\sin(\alpha)\sin(\gamma) + \sqrt{6}x\sin(\alpha)\cos(\gamma) \\
&- \cos(\alpha)\left(\sqrt{6}x\sin(\beta)\sin(\gamma) + x\sin(\beta)\cos(\gamma) + \sqrt{2}x\cos(\beta) - y\sin(\gamma) + \sqrt{6}y\cos(\gamma)\right) \\
&- \sqrt{6}y\sin(\alpha)\sin(\beta)\sin(\gamma) - y\sin(\alpha)\sin(\beta)\cos(\gamma) - \sqrt{2}y\sin(\alpha)\cos(\beta)
\end{aligned}
\tag{5.5}
$$

Now, we can move the robot by shifting its centre of mass. The control scheme to do this is shown in Figure. 5.1. Here, $\mathbf{p}_{COM_d}$ is the desired position of the centre of mass of the masses, $\mathbf{q}_d$ is the desired displacement to attain this center of mass (calculated using Equation. 5.5), $\mathbf{u}$ the control signal, and $\mathbf{q}$ and $\dot{\mathbf{q}}$ the state vector and its derivative respectively.

5.2 Trajectory tracking

5.2.1 Constant Moment Motion

At first the scheme was tested by moving the robot such that the desired position of the centre of mass, $\mathbf{p}_{COM_d}$ with respect to $\{B\}$ as $(10, 0, R)^\top$. This gives the robot a constant moment, $\mathbf{M}$, of $10mg$ along $(0, 1, 0)$. The rolling of the robot is defined by the relation:

$$\mathbf{M} = (I_{sphere} + I_{masses})\dot{\boldsymbol{\omega}} \tag{5.6}$$

where, I_{sphere} and I_{masses} are the moment of inertia due to the spherical shell and the masses respectively. For the sake of simplicity, I_{masses} is computed by assuming that the masses

are always located at the home position. However, this simplification won't hold true if the masses move far away from their home position.

The acceleration of the centre of the sphere is given as:

$$|\mathbf{a}| = |\dot{\boldsymbol{\omega}}|/R \tag{5.7}$$

The results are plotted in Figure. 5.2-5.4. As you may observe, the robot has an excellent tracking performance. This justifies the simplifications made till now.

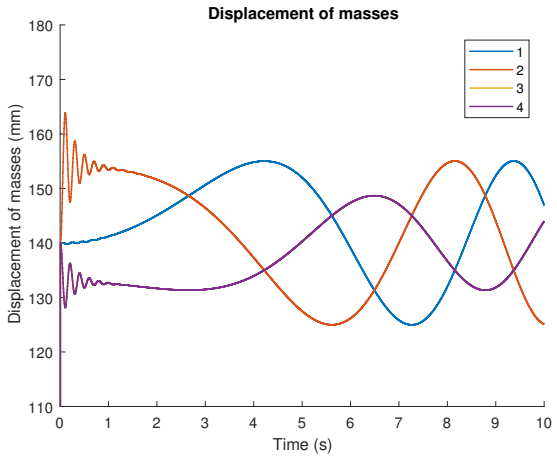


Figure 5.2: Displacement of masses: Constant moment maneuver

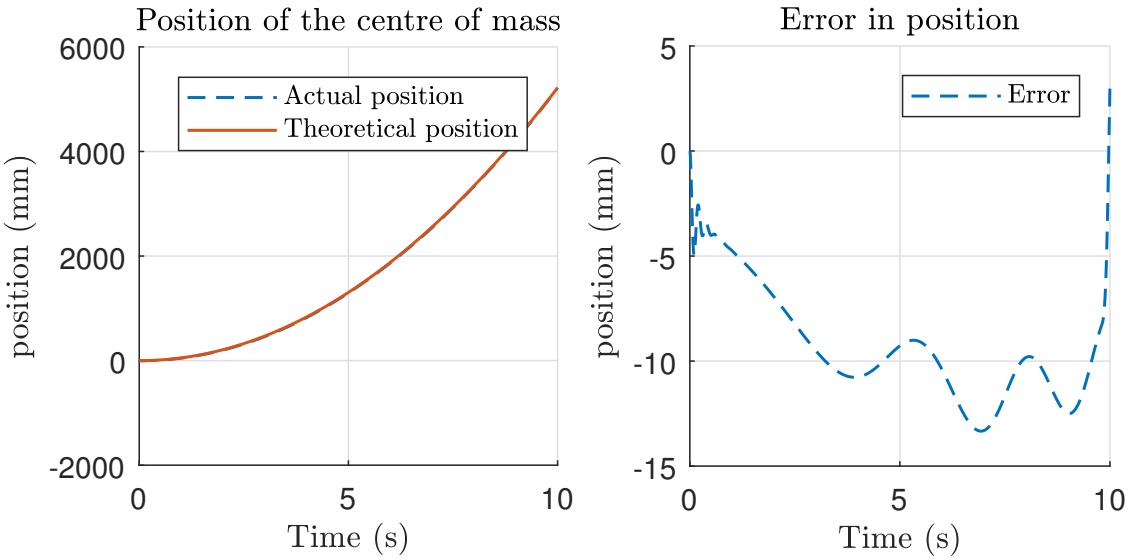


Figure 5.3: Position tracking: Constant moment maneuver

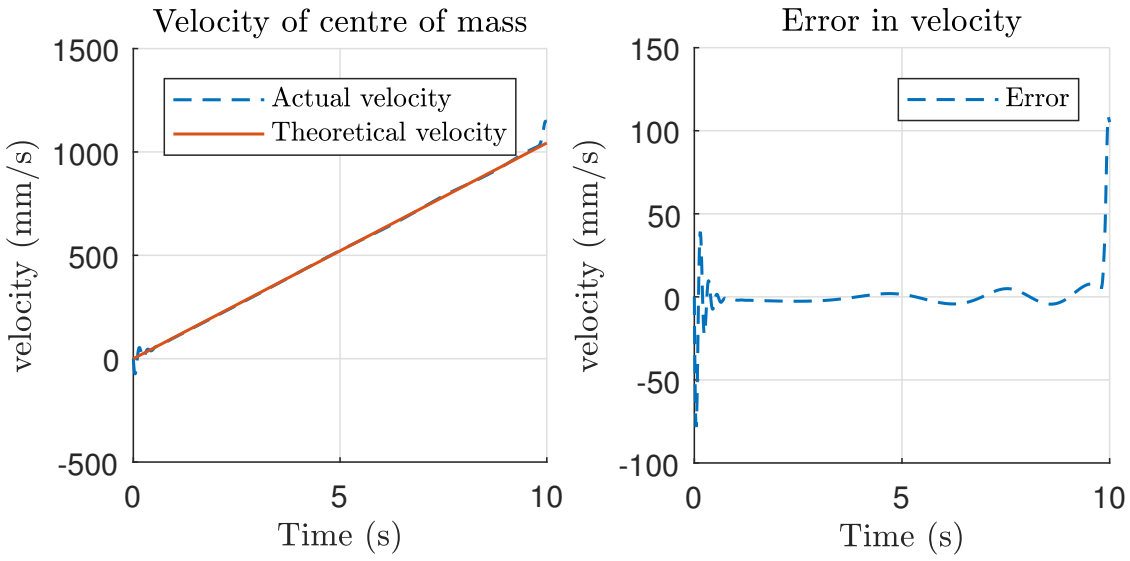


Figure 5.4: Velocity tracking: Constant moment maneuver

5.2.2 Straight Line Trajectory

In this section, we perform the trajectory tracking of the robot by making it traverse a straight line trajectory. The trajectory parameterized by a fifth order polynomial:

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{5.8}$$

with the boundary conditions,

$$\begin{aligned} q(t_i) &= q_i & q(t_f) &= q_f \\ \dot{q}(t_i) &= \dot{q}_i & \dot{q}(t_f) &= \dot{q}_f \\ \ddot{q}(t_i) &= \ddot{q}_i & \ddot{q}(t_f) &= \ddot{q}_f \end{aligned} \tag{5.9}$$

The trajectory of the robot moving in a straight line can be found from the above equation as:

$$\begin{aligned} \mathbf{p}(t) &= \mathbf{p}_0 + q(t)\mathbf{d} = \mathbf{p}_0 + (a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5)\mathbf{d} \\ \mathbf{v}(t) &= \dot{q}(t)\mathbf{d} = (a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4)\mathbf{d} \\ \mathbf{a}(t) &= \ddot{q}(t)\mathbf{d} = (2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3)\mathbf{d} \end{aligned} \tag{5.10}$$

where, $\mathbf{p}_0$ is the starting point and $\mathbf{d}$ the direction of motion.

The trajectory control scheme is shown in Figure. 5.5. The desired position ($\mathbf{p}_d$), velocity ($\dot{\mathbf{p}}_d$), and acceleration ($\ddot{\mathbf{p}}_d$) of the centre of the robot is generated by the trajectory generator. The trajectory controller gives a control signal, $\mathbf{a}_d$, defined as:

$$\mathbf{a}_d = \ddot{\mathbf{p}}_d + K_d(\dot{\mathbf{p}}_d - \dot{\mathbf{p}}_d) + K_p(\mathbf{p}_d - \mathbf{p}) \tag{5.11}$$

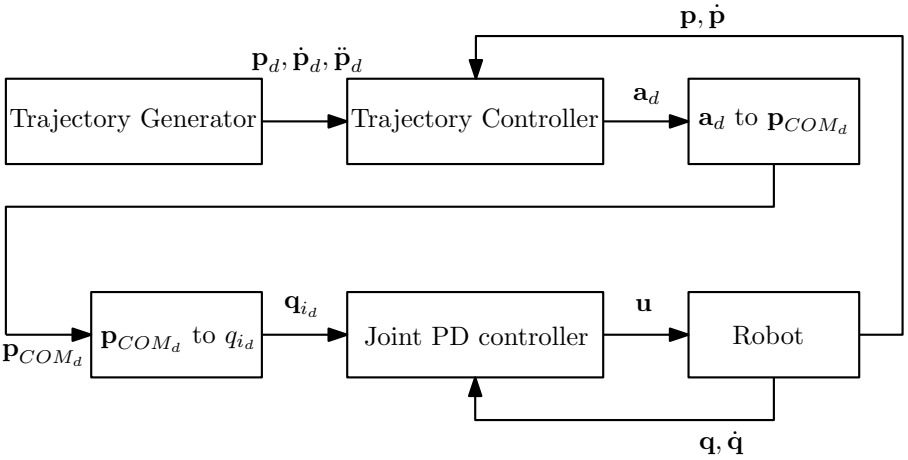


Figure 5.5: Trajectory control scheme

This signal is then converted to $\mathbf{p}_{COM_d}$ to control the robot as explained in the previous section.

Now we try to move the in a straight line, starting from $(0,0)^\top$, ending at $(0,1000)^\top$, with zero initial and final velocities and accelerations. Therefore,

$$\begin{aligned} q(t_i) &= 0 & q(t_f) &= 1000 \\ \dot{q}(t_i) &= 0 & \dot{q}(t_f) &= 0 \\ \ddot{q}(t_i) &= 0 & \ddot{q}(t_f) &= 0 \\ \mathbf{p}_0 &= (0,0)^\top & \mathbf{d} &= (0,1)^\top \end{aligned} \tag{5.12}$$

As shown in Figure. 5.6 - 5.7, the robot was able to successfully track the desired trajectory with a tracking error of the order of 0.2%.

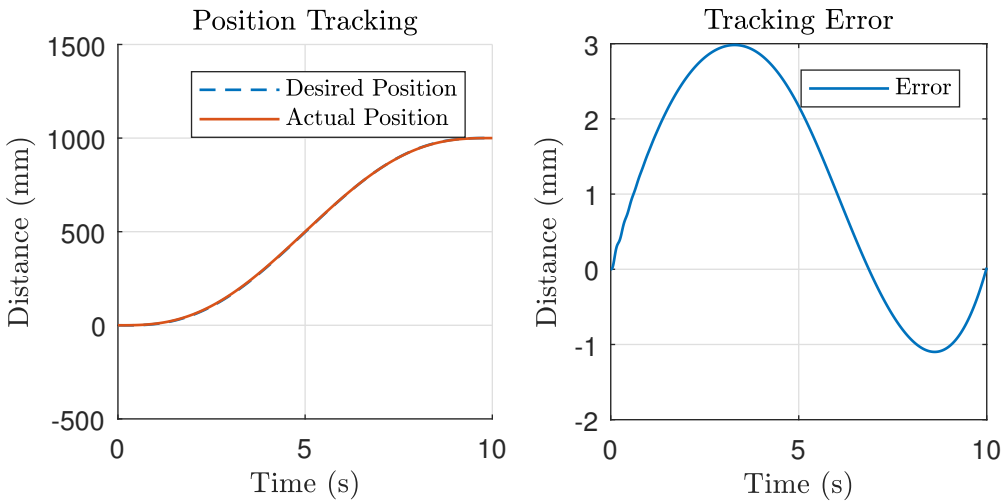


Figure 5.6: Position tracking: Straight line trajectory

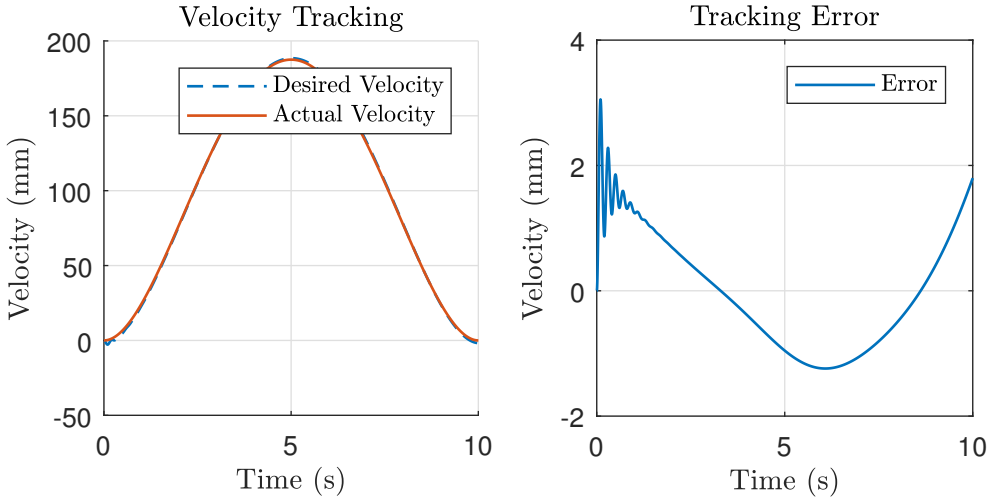


Figure 5.7: Velocity tracking: Straight line trajectory

Chapter 6

Conclusion and Future Works

The aim of the project was to develop the analytical model of the 3D rolling robot with mobile masses and VSS actuators in parallel. This is a preliminary to studying the effect of VSS on energy consumption. At first, a literature review of the state of the art design of rolling robots and VSS actuators was performed. Subsequently, the dynamic models of three reciprocating-mass-type rolling robots; (i) without elastic elements (nominal), (ii) with fixed stiffness springs in parallel, and (iii) with VSS actuators in parallel were developed in `MATLAB`.

The dynamic models were validated in `MSC ADAMS` by performing different set of experiments. The error between the analytical model and the `ADAMS` model was found to be of the magnitude 10^{-6}. This indicates the accuracy of the model.

We were also able to implement a simple control algorithm that is able to steer the spherical robot by continuously changing the location of its centre of mass. A closed loop control between `Simulink` and `ADAMS/Controls` was used to test the control algorithm.

The developed model can be used to study the effect of variable stiffness springs in reducing the input torques of the motors that actuate the mobile masses. At a later stage, a prototype can be developed to test the robot in real world applications.

Bibliography

[1] Rafael Balderas Hill, Sébastien Briot, Abdelhamid Chriette, and Philippe Martinet. Minimizing input torques of a high-speed five-bar mechanism by using variable stiffness springs. In Vigen Arakelian and Philippe Wenger, editors, *ROMANSY 22 – Robot Design, Dynamics and Control*, pages 61–68, Cham, 2019. Springer International Publishing.

[2] M. Furet. *Design and control of exploration robot with reduced energy consumption.* Master thesis, Ecole Centrale de Nantes, 2017.

[3] Werner Schiehlen and Nils Guse. Control of limit cycle oscillations. In G. Rega and F. Vestroni, editors, *IUTAM Symposium on Chaotic Dynamics and Control of Systems and Processes in Mechanics*, pages 429–439, Dordrecht, 2005. Springer Netherlands.

[4] HongChao Zhuang, HaiBo Gao, ZongQuan Deng, Liang Ding, and Zhen Liu. A review of heavy-duty legged robots. *Science China Technological Sciences*, 57(2):298–314, Feb 2014.

[5] Filip Tomik, Shahin Nudehi, Louis L Flynn, and Ranjan Mukherjee. Design, fabrication and control of spherobot: A spherical mobile robot. *Journal of Intelligent & Robotic Systems*, 67(2):117–131, 2012.

[6] Richard Chase and Abhilash Pandya. A review of active mechanical driving principles of spherical robots. *Robotics*, 1(1):3–23, 2012.

[7] A. Halme, T. Schonberg, and Yan Wang. Motion control of a spherical mobile robot. In *Advanced Motion Control, 1996. AMC '96-MIE. Proceedings., 1996 4th International Workshop on*, volume 1, pages 259–264 vol.1, Mar 1996.

[8] 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*, 217(6):457–467, 2003.

[9] C. Camicia, F. Conticelli, and A. Bicchi. Nonholonomic kinematics and dynamics of the sphericle. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, volume 1, pages 805–810 vol.1, 2000.

[10] A. Bicchi, A. Balluchi, D. Prattichizzo, and A. Gorelli. Introducing the sphericle: an experimental testbed for research and teaching in nonholonomy. In *Proceedings of International Conference on Robotics and Automation*, volume 3, pages 2620–2625 vol.3, Apr 1997.

[11] Francois Michaud and Serge Caron. Roball, the rolling robot. *Autonomous robots*, 12(2):211–222, 2002.

[12] Bo Zhao, Pengfei Wang, Haiyan Hu, Mantian Li, and Lining Sun. Study on turning in place of a spherical robot based on stick-slip principle. In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pages 771–775. IEEE, 2009.

[13] Joong-Cheol Yoon, Sung-Su Ahn, and Yun-Jung Lee. Spherical robot with new type of two-pendulum driving mechanism. In *Intelligent Engineering Systems (INES), 2011 15th IEEE International Conference on*, pages 275–279. IEEE, 2011.

[14] Puyan Mojabi et al. Introducing august: a novel strategy for an omnidirectional spherical rolling robot. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 3527–3533. IEEE, 2002.

[15] Y. Sugiyama, A. Shiotsu, M. Yamanaka, and S. Hirai. Circular/spherical robots for crawling and jumping. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3595–3600, April 2005.

[16] H. B. J. Brown and Yangsheng Xu. A single wheel, gyroscopically stabilized robot. *IEEE Robotics Automation Magazine*, 4(3):39–44, Sept 1997.

[17] J. Qingxuan, Z. Yili, S. Hanxu, C. Hongyu, and L. Hongyi. Motion control of a novel spherical robot equipped with a flywheel. In *2009 International Conference on Information and Automation*, pages 893–898, June 2009.

[18] Antonio Bicchi, Giovanni Tonietti, Michele Bavaro, and Marco Piccigallo. Variable stiffness actuators for fast and safe motion control. In *Robotics Research. The Eleventh International Symposium*, pages 527–536. Springer, 2005.

[19] Bram Vanderborght, Alin Albu-Schäffer, Antonio Bicchi, Etienne Burdet, Darwin G Caldwell, Raffaella Carloni, MG Catalano, Oliver Eiberger, Werner Friedl, Ganesh Ganesh, et al. Variable impedance actuators: A review. *Robotics and autonomous systems*, 61(12):1601–1614, 2013.

[20] Tom Verstraten, Philipp Beckerle, Raphaël Furnémont, Glenn Mathijssen, Bram Vanderborght, and Dirk Lefeber. Series and parallel elastic actuation: Impact of natural dynamics on power and energy consumption. *Mechanism and Machine Theory*, 102:232–246, 2016.

Appendices

1 ADAMS

1.1 Forces (Contact Friction)

To get a satisfactory representation of the ADAMS model of the rolling robot in the real world environment it is important to set the contact forces between the sphere and the base accurately.

ADAMS has mainly 3 built-in contact methods: the *IMPACT* function model, the *POISSON* restitution model and the *COULOMB* friction model. Generally, when the contact is not strictly elastic and the frictional force cannot be neglected, one can select the *IMPACT* function in the Normal Force field and *COULOMB* function in the Friction Force field.

In the *IMPACT* function model, four variables must be defined: Stiffness, Force Exponent, Damping and Penetration Depth. These correspond to the four arguments in the actual *IMPACT* function.

$$IMPACT(x, \dot{x}, x_1, k, e, c_{max}, d) \tag{1}$$

Table 1: ADAMS *IMPACT* model parameters

x	An expression that specifies a distance variable used to compute the IMPACT function
$\dot{x}$	An expression that specifies the time derivative of x to *IMPACT*
x_1	A positive real variable that specifies the free length of x. If x is less than x_1, then ADAMS calculates a positive value for the force. Otherwise, the force is zero
k	A non-negative real variable that specifies the stiffness of the boundary surface interaction
e	A positive real variable that specifies the exponent of the force deformation characteristic.
c_{max}	A non-negative real variable that specifies the maximum damping coefficient
d	A positive real variable that specifies the boundary penetration at which ADAMS applies full damping

The arguments $x, \dot{x}$ and x_1 are geometry-related expressions and are calculated during run-time. The *IMPACT* function is given below in equation 2.

$$F = \begin{cases} 0 & if\ x > x_1 \\ k(x_1 - x)^e - c_{max}\dot{x} * STEP(x, x_1 - d, 1, x_1, 0) & if\ x \leq x_1 \end{cases} \tag{2}$$

It cab be seen from equation 2 that the force becomes non-zero when the distance between the two objects is smaller than the free length of x.

- The contact stiffness k can be found out using the Hertzian Contact theory. However the magnitude of k is generally quite high in the range of 10^{7-11}, which means it can be approximated to a reasonably high value. We set k equal to 1×10^8.

- The force exponent e expresses the collision area change due to the penetration and is given as $e = 3/2$ in the Hertz contact model. Nevertheless, the recommended values of e are in the range of 1.5 to 2.2, using the lower values for soft metals and the higher ones for stiff ones. We set the force exponent $e = 2.2$ assuming a material of high stiffness.

- Hertzian contact theory states that at contact, the object deformation dissipates energy from the system. `ADAMS` uses a damping parameter(c_{max}) to create a damping force that dissipates energy from the system. Since the dissipation of energy depends on the contact area and contact stiffness, the damping value in the *IMPACT* function is recommended to be a small fraction of the stiffness value: $C_{max} < 0.01k$. So we set damping $c_{max} = 1 \times 10^4$.

- The penetration depth d is the measure of how the damping coefficient ramps up from zero to c_{max}. This value should be smaller than the maximum penetration depth and accordingly we set $d = 1 \times 10^{-4}$.

In the *COULOMB* friction model, there are four parameters to be set: static coefficient of friction μ_s, dynamic coefficient of friction μ_d, the stiction and friction transition velocities V_s and V_d. As we know the transition from static to dynamic friction when there is a relative velocity between the two bodies, is a discontinuity. However `ADAMS` models this discontinuity as a continuous cubic step function as can be seen in figure 1.

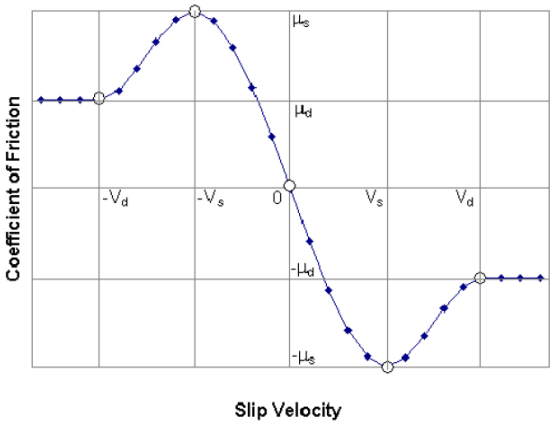


Figure 1: `ADAMS` Coefficient of Friction varying with Slip Velocity

- As can be seen from the figure 1, when the contact velocity is between zero and V_s, the coefficient of friction increases from zero to μ_s. For contact velocities between V_s and V_d, the coefficient of friction decreases from μ_s to μ_d. The coefficients μ_s and μ_d

depends on the material of the two bodies in contact. We can choose the coefficients based on known material properties, however we chose to keep the ADAMS default value of $\mu_s = 0.3$ and $\mu_d - 0.1$. Note that $\mu_s > \mu_d$ when choosing your own values.

- The discontinuity at zero velocity is spread over a velocity range between zero and V_s in the ADAMS Coulomb model, making the coefficient of friction a continuous function of the contact velocity. When the contact velocity is equal to or greater than V_d, the coefficient of friction remains constant at μ_d. We keep the default values in ADAMS, $V_s = 0.1$ and $V_d = 1.0$. However the user is free to chose the transition velocities, with V_s being close to zero and smaller than V_d.

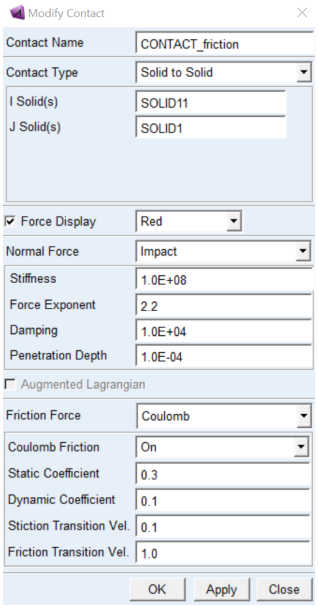


Figure 2: ADAMS Contact Friction Model

1.2 System and Data Elements

A brief description of the state variables is given below in table 2.

Table 2: ADAMS System Elements

q_i_out	The displacement of Mass i $(i = 1, 2, 3, 4)$ from the core (output)
dq_i_out	The Velocity of Mass i $(i = 1, 2, 3, 4)$ from the core (output)
ddq_i_out	The acceleration of Mass i $(i = 1, 2, 3, 4)$ from the core (output)
$roll$	Roll (Euler angle) of the sphere with respect to ground (output)
$Pitch$	Pitch (Euler angle) of the sphere with respect to ground (output)
Yaw	Yaw (Euler angle) of the sphere with respect to ground (output)
W_i_out	Angular velocity of the sphere with respect to the ground about the i axis $(i = X, Y, Z)$ of the ground (output)
dW_i_out	Angular acceleration of the sphere with respect to the ground about the i axis $(i = X, Y, Z)$ of the ground (output)
Pos_{ij}_wrtg	This output state variable gives the position of the i^{th} Mass $(i = 1, 2, 3, 4)$ with respect to the ground along the ground j axis $(j = X, Y, Z)$
Vel_{ij}_wrtg	This output state variable gives the velocity of the i^{th} Mass $(i = 1, 2, 3, 4)$ with respect to the ground along the ground j axis $(j = X, Y, Z)$
Acc_{ij}_wrtg	This output state variable gives the acceleration of the i^{th} Mass $(i = 1, 2, 3, 4)$ with respect to the ground along the ground j axis $(j = X, Y, Z)$
$Core\_X\_wrtg$ $Core\_Y\_wrtg$	This output state variable gives the position of the sphere with respect to the ground about the X and Y axis
$Core\_Xd\_wrtg$ $Core\_Yd\_wrtg$	This output state variable gives the velocity of the sphere with respect to the ground about the X and Y axis
F_mass_i	The force on Mass i along the Z axis of the center of mass marker which is along the spoke.
qs_i_out	Spring base $(i = 1, 2, 3, 4)$ displacement from the core.