Mars Sample-Return Rover

ENPM 662 Final Project

Robert Vandemark Diane Ngo

December 2020

Contents

1 Abstract **2**

2 Introduction **3**

2.1 Motivation 4

2.2 Assumptions 4

3 Robot Design **5**

4 Forward Kinematics **7**

4.1 Vehicle 7

4.2 Arm 7

5 Velocity Kinematics **12**

5.1 Calculate Displacements 13

5.2 Calculate Wheel Velocities and Estimated Elapsed Time 15

6 Inverse Kinematics **17**

6.1 Arm 17

6.2 Combined 20

7 Scope of Achievement **21**

8 Model Validation and Testing **22**

9 Conclusion **23**

Bibliography **24**

1 Abstract

Space exploration is still a topic of interest because technology is continuously developing. Exploring and researching Mars is the most convenient way to trial and test new space technologies since it is the closest planet to Earth that is able to conduct research on. NASA approached Mars exploration through the Sample-Return Rover, a robot that can explore unknown terrain autonomously and be able to collect samples. The earliest primitive model was launched in 1997, and the latest one being 2020 with the Curiosity Rover. This project focuses on recreating the earlier version of the rover in SolidWorks and being able to simulate the exploration in a 3D environment with Gazebo.

2 Introduction

This project focuses on the Mars *Sample-Return Rover* (SRR), as seen in Figure 2.1, developed by NASA's Jet Propulsion Laboratory (JPL). Exploring the unknown is fascinating with the challenges that are present. Robotics is a field suitable to approach these challenges because they have the capability to operate autonomously. The model will be created in Solidworks and then simulated in Gazebo. The report explains the elements of the project, starting with the rover's geometric aspects such as its mechanical design, joints, links, and arm. The rover consists of the vehicle base with the chassis and wheels, and an attached *Four Degrees of Freedom* (4DOF) arm.

The rover's chassis has an interesting structure with the front and rear links are connected with one joint, but have independent movement. With a total range of 180 degrees, the rover can scale steep cliffs because of the design. The front wheels can also rotate side-to-side so that the rover can turn. Attached to the chassis is a 4DOF arm. It rotates at its base, an elbow joint, and the wrist joint can rotate sideways and and up and down. The end-effector on the arm allows for sample-collection.

With the challenges of unknown terrain and inability to remote control the robot, extensive research and testing has to be made before launching the robot. The scope of this project focuses on the Sample-Return Rover. Model assumptions will be discussed. Afterwards, the design of the robot will be explained with analysis of the kinematics of the vehicle and the arm.

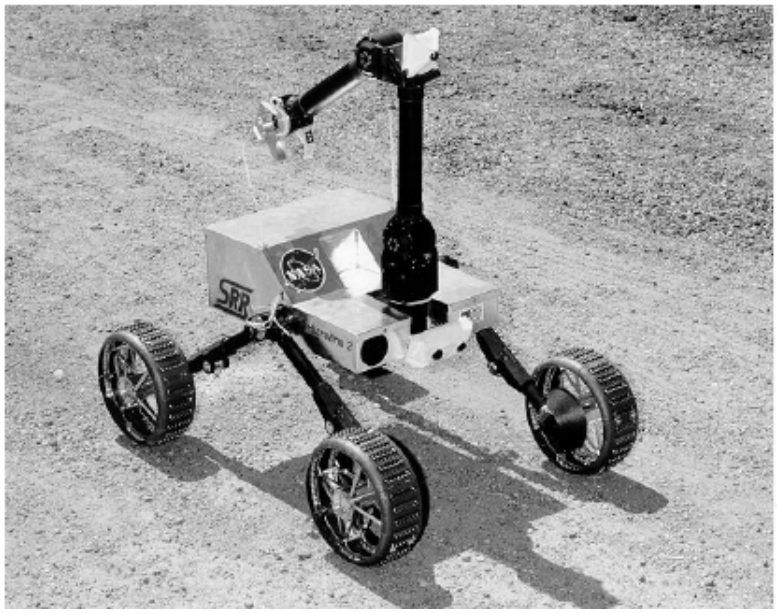

Figure 2.1: Early Model of the Sample Return Rover

Motivation

Space robotics is a big area of interest. Not only has there been the Sample-Return Rover, but there is also the CubeSat, BioSentinel, and many others. The Sample-Return Rover is a great choice for this project as the information gained from the class will be used on the rover. There are many possible usages for space robotics. They can serve the purpose of exploring unknown territory, surveillance of the Earth, or maintenance on existing robots/satellites.

Each different type of robot will deal with their own challenges, whether it is dealing with low or zero gravity, extreme temperature changes, or rocky terrain. It is amazing how diverse space robotics can be. The area of research is vast and takes much effort to be in a robotics field. The intention of this project is to get an introductory hands-on experience with both mobile robots and a manipulator simulation. Calculating forward, inverse, and velocity kinematics are a big topic for this project, as well as implementation of those calculations into algorithms.

Assumptions

Since the scope of this project deals with simulating a robot in outer space, there are assumptions to be made because certain things cannot be measured easily. To deal with this issue and other design factors, there are assumptions to follow.

1. The robot is a rigid body, any torque or force acted upon any link will not be deformed.
2. The steering design is based off the Ackermann-Steering model.
3. All motion or revolution will be around the z-axis.
4. The simulations will be in Gazebo and RVIZ.

3 Robot Design

The robot base comprises of four legs, with two on each side that are concentric. On either side, two of the legs are connected via a revolute joint and is concentric. The front wheels have a revolute joint connected to the wheels so that the robot can steer left or right. To have the robot at the same height, the rear wheels have the same joint however it is fixed. The robot arm that is attached has four degrees of freedom.

Modeling the robot is split into the vehicle and arm separately. With this method, both models can be configured and tested properly in Gazebo in order to use tele-op on individual joints. Figure 3 shows the rover modeled in SolidWorks. Table 3.1 lists the dimensions of the robot and links. Figure 3 and Figure 3 are images of the arm for reference.

Link	*Length*
Chassis Length	18in
Chassis Width	12in
Chassis Max Height	7in
Leg from Center Joint	8.6in
Length of Steering Link	5.25in
Wheel Diameter	7.5in
Wheel Track	19.3in
Wheelbase	13.9in

Table 3.1: Table of Dimensions of the Rover

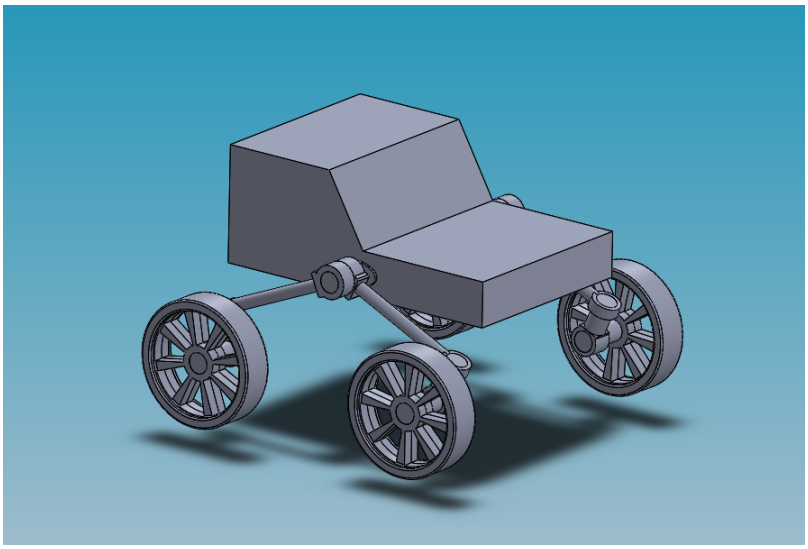

Figure 3.1: Sample Return Rover in SolidWorks

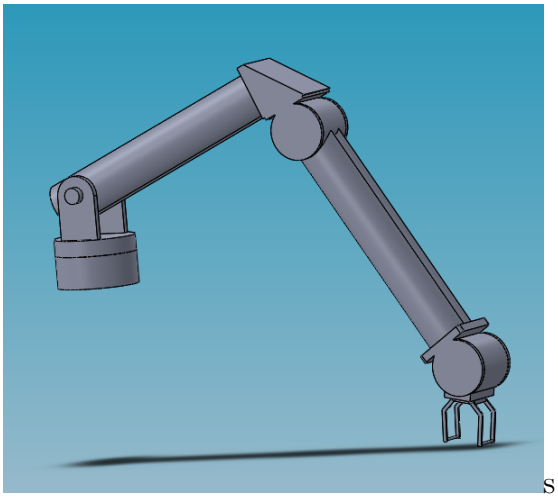 s

Figure 3.2: Arm Model in SolidWorks

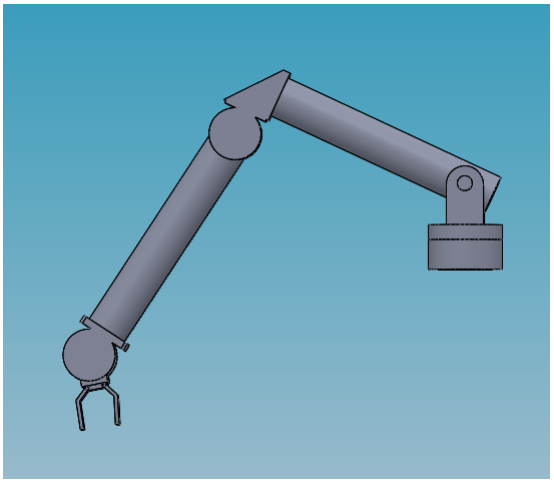

Figure 3.3: Side View of Robot Arm

4 Forward Kinematics

Vehicle

The rover uses an Ackermann steering mechanism to be able to navigate, like that seen in Figure 4.1. Unlike the average robotic arm, it is not a serial connection. Forward kinematics cannot be used on a mobile robot. Ackermann steering allows the front wheels to rotate independently, about a common center point. This center point is called the *Instantaneous Center of Curvature* (ICC). In general, it uses a four bar linkage in a trapezoidal shape. However the Curiosity Mars Rover and the Sample-Return Rover does not use this linkage, but the model still holds.

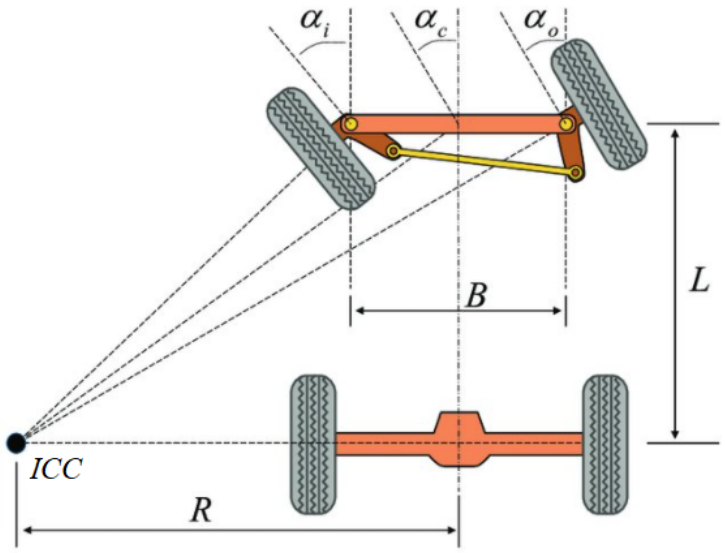

Figure 4.1: Diagram for Ackermann Steering

Arm

In robotics, forward kinematics is used to find the position and orientation of the robot's end-effector (or gripper) given the joint angles. In this course, joints can have two different types: *revolute* and *prismatic*. A revolute joint rotates about an axis, while a prismatic joint translates linearly along an axis. In serial kinematic chains, such as the arm seen in Figure 3, each coordinate frame assigned to the distal end of a link i is dependent on the position and geometry of the $(i-1)^{th}$ joint and link, respectively. The frame i resolved in frame i - *1* represents a combination of a rotation performed by the $(i-1)^{th}$ joint with a successive translation down the length of link i - *1*. Chaining these homogeneous transformation matrices together results in the transformation of the end-effector in the

robot's base frame.

The robot arm itself is 4DOF where each joint is revolute. However, as seen in Figure 4.2, there are eight frames in total, because four dummy frames are for Gazebo to properly render in the arm. Its corresponding DH table is shown in Table 4.2, and the values for the nonzero static lengths can be seen in Table 4.2. A list of the links are shown in Figure 4.2. The links *LeftFinger* and *RightFinger* are just there for Gazebo to be able to rotate the end-effector fingers, but they are not used in the DH parameters and table.

Note that the arm's home position starts with all angles being at 0, so the arm is fully upward. Each joint has a limit for its range of motion so that the robot does not crash into itself. Equation 4.1 to 4.4 offer reasonable ranges of allowable motion.

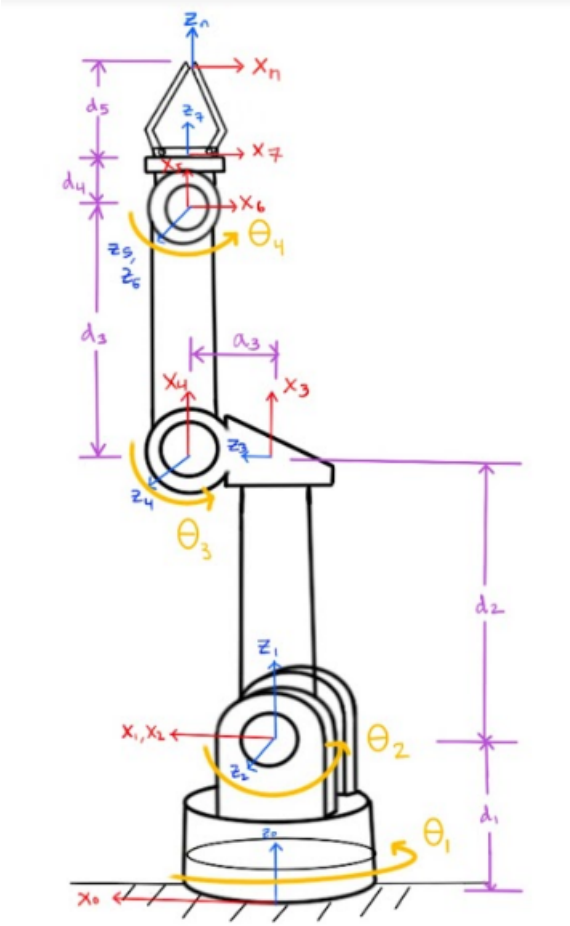

Figure 4.2: DH Frames of the Arm

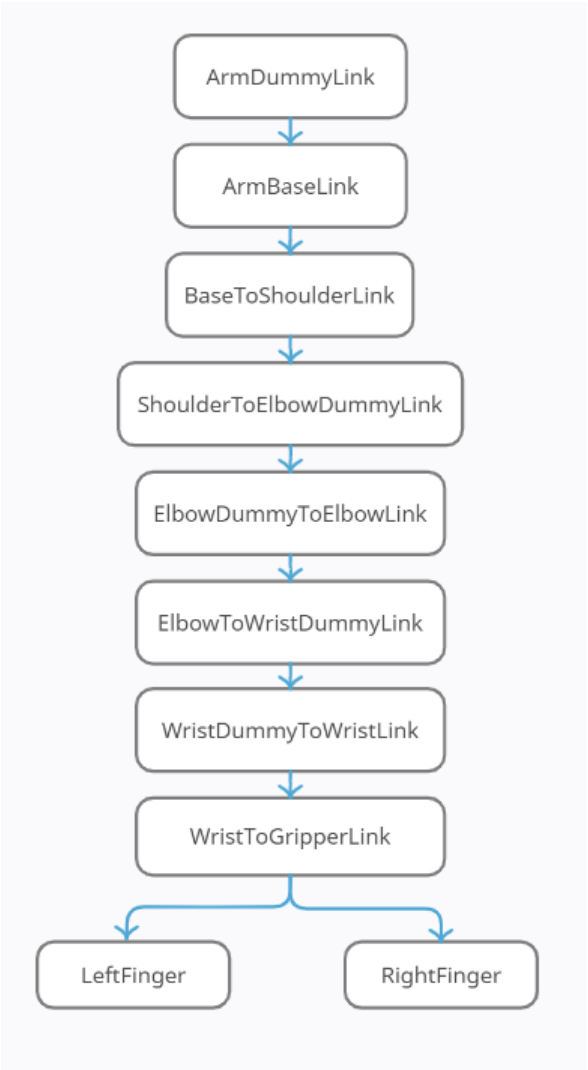

Figure 4.3: Flowchart of the Arm Links

Link i	θ	d	α	a
1	θ_1	d_1	0	0
2	0	0	-90	0
3	θ_2 - 90	0	-90	d_2
4	0	a_3	90	0
5	θ_3	0	0	d_3
6	θ_4 - 90	0	-90	0
7	0	d_4	0	0
n	0	d_5	0	0

Table 4.1: DH Table for the Arm

Distance between Frames	*Length*
Base to Frame 1 (d_1)	2.9in
Frame 2 to Frame 3 (d_2)	7.73in
Frame 3 to Frame 4 (a_3)	1.76in
Frame 4 to Frame 5 (d_3)	8.93in
Frame 6 to Frame 7 (d_4)	1in
Frame 7 to Frame n (d_5)	1.41in

Table 4.2: Table of Link Lengths of the Arm

$$\theta_1 = [-\pi,\ +\pi] \tag{4.1}$$

$$\theta_2 = \left[0,\ +\frac{2\pi}{3}\right] \tag{4.2}$$

$$\theta_3 = [0,\ +\pi] \tag{4.3}$$

$$\theta_4 = \left[-\frac{2\pi}{3},\ +\frac{2\pi}{3}\right] \tag{4.4}$$

One method to start calculating the forward kinematics is to create DH matrices T_n^{n-1}, which are composed of rotations about the z-axis, translations in the z-axis, rotations about the x-axis, and finally translations in the x-axis. The basic rotation matrix for a rotation about the z-axis with an angle θ can be shown in Equation 4.5, and Equation 4.6 for rotation about the x-axis by angle α. The translation vectors can be seen in Equations 4.8 and 4.7 for along the z and x axes by a displacement d and a, respectively. Using the DH Table, the homogeneous transformation matrices can now be written. The setup for this matrix is seen in Equation 4.9.

$$R_{z,\theta} = \begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

$$R_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix} \tag{4.6}$$

$$t_{n_z}^{n-1} = \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} \tag{4.7}$$

$$t_{n_x}^{n-1} = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix} \tag{4.8}$$

$$H_n^0 = \begin{bmatrix} R_n^0 & t_n^0 \\ 0 & 1 \end{bmatrix} \tag{4.9}$$

Another method is to observe the geometry of the workspace, which is the set of positions and orientations that the end effector can accomplish, and find the restrictions created by it. This type of approach was taken for this project to solve for the four dimensions of actuation, the 3D position $[x_n^0\ y_n^0\ z_n^0]^T$ and a rotation off the z-axis of the robot's shoulder joint γ.

The base joint is the only one that rotates about the base's z-axis, so the values for x_n^0 and y_n^0 can be thought of as a projection of the arm on to the (x_0,y_0) plane, where the length of this projection is a function of the shoulder, elbow, and wrist joints, as well as their child links. This length rotates about the z axis with the base joint. This geometry describes how to derive x_n^0 and y_n^0, seen in Equations 4.10 and 4.11. The z coordinate by definition is the displacement of linkage in the z-axis. Considering the fact that the rest of the three joints act in the xz plane, the solution can be solved as a summation of the length of each link that traverses its local x-axis times the cosine of the net angle and the length of each link that traverses its local z-axis times the sine of the net angle. For this arm, that results in Equation 4.12, as θ_1 can not cause actuation in the z-axis. Finally, because γ is essentially the angle between the xy plane of the base and the surface that the vehicle is on, assuming that the vehicle is flat relative to the surface, then γ can be solved with Equation 4.13.

$$x_n^0 = c_1\left(d_2 s_2 + a_2 c_2 + d_3 s_{23} + (d_4 + d_5) s_{234}\right) \tag{4.10}$$

$$y_n^0 = s_1\left(d_2 s_2 + a_2 c_2 + d_3 s_{23} + (d_4 + d_5) s_{234}\right) \tag{4.11}$$

$$z_n^0 = d_1 + d_2 c_2 - a_2 s_2 + d_3 c_{23} + (d_4 + d_5) c_{234} \tag{4.12}$$

$$\gamma = \theta_2 + \theta_3 + \theta_4 \tag{4.13}$$

Where:

$$c_{ij\ldots} = cos(\theta_i + \theta_j + \ldots) \tag{4.14}$$

$$s_{ij\ldots} = sin(\theta_i + \theta_j + \ldots) \tag{4.15}$$

5 Velocity Kinematics

As described earlier, moving the arm's end effector to a desired position in space relative to the world frame is a forward kinematics problem. However, for a vehicle, moving to a position in space is primarily a velocity kinematics problem, because the velocity of the vehicle body has to be maintained over a period of time, which implies that the speeds for the wheels have to be coordinated to follow a specified path.

While this vehicle is technically a *Five Degrees of Freedom* (5DOF) vehicle, because the chassis can rotate about both its long and short axes with the joints attached to its legs, its basic pose can be described as an x and y coordinate pair with an orientation that the vehicle is facing θ. Movement between an initial pose $[x_i \ y_i \ \theta_i]$ and final pose $[x_f \ y_f \ \theta_f]$ such as this can be thought of an initial turn, a straight movement, and then a final turn to get to the desired orientation. A top view example of this geometry can be seen in Figure 5.1. Each set of concentric circles is an illustration of the path that a vehicle following an Ackerman model takes. The inner ring is the path for the inside wheels, the middle ring is the path of the lateral center of the chassis, and the outer ring is the path of the outer wheels.

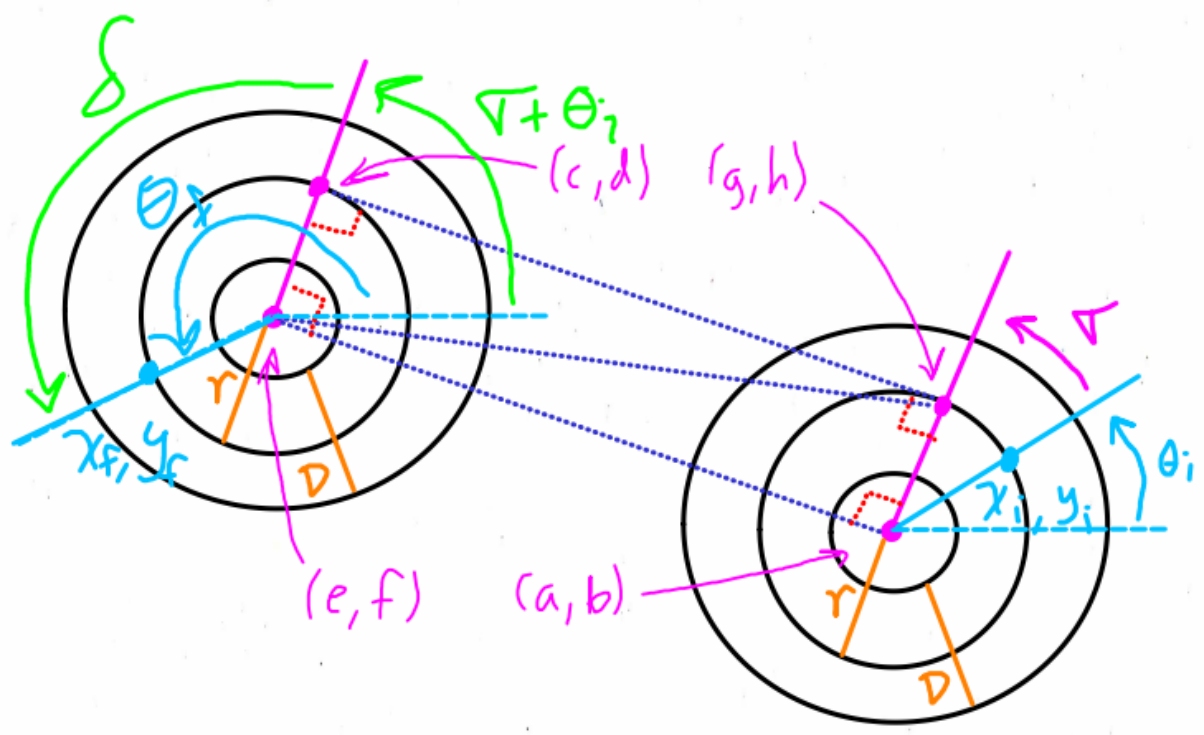

Figure 5.1: Overhead View of Example Vehicle Velocity Kinematics Problem

Calculate Displacements

First, the distances traveled in each of these three movement segments need to be calculated. These are the arc of the first turn σ, the length of the linear movement L between (g,h) and (c,d), and then the arc of the second turn δ. To simplify the problem, the steering angle for the inner wheel is assumed to be the sharpest turn allowed, which means the distance between the *Instantaneous Center of Curvature* (ICC) and the later center of the vehicle is known. This allows for the values of a, b, e, and f to be calculated in terms of the initial and final positions of the vehicle, according to Equations 5.1 to 5.4.

$$a = x_i - rc_i \tag{5.1}$$

$$b = y_i - rs_i \tag{5.2}$$

$$e = x_f - rc_f \tag{5.3}$$

$$f = y_f - rs_f \tag{5.4}$$

Where again:

$$c_{i,f} = cos(\theta_{i,f}),\ s_{i,f} = sin(\theta_{i,f}) \tag{5.5}$$

Now consider the geometry of the triangle formed by points (e,f), (a,b), and (g,h), as seen in Figure 5.2. Solving for the position g would then allow to solve for the initial arc of the vehicle's movement σ, as seen in Equation 5.6, and can be solved after finding the value for the angle β.

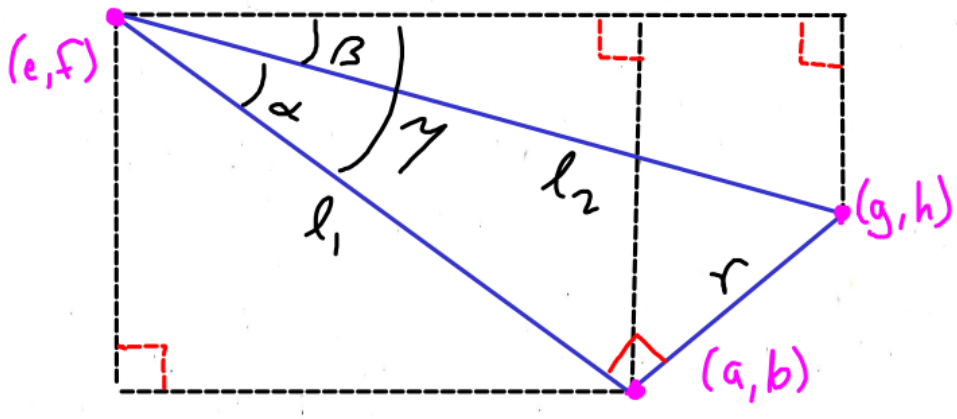

Figure 5.2: Triangle Formed by Velocity Kinematics Coordinates of Interest

$$\begin{aligned} g &= a + rc_{\sigma\theta_i} \\ &= x_i - rc_i + rc_{\sigma\theta_i} \end{aligned} \tag{5.6}$$

β can be calculated according to Equation 5.7:

$$\beta = \gamma - \alpha \tag{5.7}$$

Therefore, the values of γ and α must be solved for. Equation 5.8 describes how γ can be easily calculated:

$$\gamma = atan2(f - b,\ a - e) \tag{5.8}$$

Given Equations 5.1 to 5.4, the distance l_1 is known according to Equation 5.9. This can be used to calculate α, as seen in Equation 5.10.

$$l_1 = \sqrt{(a-e)^2 + (f-b)^2} \tag{5.9}$$

$$\alpha = -atan\left(\frac{r}{l_1}\right) \tag{5.10}$$

Substituting Equations 5.8 and 5.10 into 5.7 results in Equation 5.11.

$$\beta = atan2(f - b,\ a - e) + atan\left(\frac{r}{l_1}\right) \tag{5.11}$$

Because the values of α and l_1 are known, l_2 can be calculated as such:

$$l_2 = \sqrt{l_1^2 + r^2} \tag{5.12}$$

Finally, g can be put into terms of β geometrically and simplified using Equation 5.3, according to Equation 5.13.

$$\begin{aligned} g &= e + l_2 c_\beta \\ &= x_f - r c_f + l_2 c_\beta \end{aligned} \tag{5.13}$$

Combining Equations 5.6 and 5.13 yields Equation 5.14 and can be rearranged to solve for σ according to 5.15. The relationship between δ and σ is straightforward, as seen in Equation 5.14

$$x_i - r c_i + r c_{\sigma\theta_i} = x_f - r c_f + l_2 c_\beta \tag{5.14}$$

$$\sigma = acos\left(\frac{x_f - x_i + l_2 c_\beta}{r} + c_i - c_f\right) - \theta_i \tag{5.15}$$

$$\delta = \theta_f - \theta_i - \sigma \tag{5.16}$$

Finally, L has already implicitly been solved for, as the distance between (e,f) and (a,b) is the same as that of (c,d) and (g,h). Therefore:

$$L = l_1 \tag{5.17}$$

Calculate Wheel Velocities and Estimated Elapsed Time

Now that the values of σ, L, and δ are known, and while utilizing a simple assumption, all of the wheel velocities during the three movement segments, and the time each of those moves would ideally take, can be calculated. The arc length for the vehicle's outer wheels during the first turn, S_{oft}, can be calculated according to Equation 5.18.

$$S_{oft} = R_w \Delta\omega_{oft} \tag{5.18}$$

Where R_w is the radius of the vehicle's wheels. The total rotation for the outside wheels $\Delta\omega_{oft}$ can also be calculated as the wheel's angular velocity over a period of time, according to Equation 5.19.

$$\Delta\omega_{oft} = \dot{\omega_{oft}} \Delta t_{ft} \tag{5.19}$$

The subscript for Δt_{ft} does not include the 'o' because the elapsed time for the outside wheels is the same as the inside wheels over the course of the turn, which is a necessary conclusion to utilize later.

The aforementioned simple assumption is that the outside wheels during a turn will move at the peak velocity, which is a guaranteed safe assumption according to the Ackerman steering model, because the wheels must rotate slower to maintain a constant ICC. Therefore:

$$\dot{\omega} = \dot{\omega}_{max} \tag{5.20}$$

Rearranging the combination of Equations 5.18 to 5.20 to solve for the elapsed time yields Equation 5.21.

$$\Delta t_{ft} = \frac{S_{oft}}{R_w \cdot \dot{\omega}_{max}} \tag{5.21}$$

Equations 5.22 and 5.23 are the same as the outside wheels except to reflect the inside wheels' movement. This time, Δt_{ft} is not solved for, but is utilized to solve for the inside wheels' angular velocities, according to Equation 5.24.

$$S_{ift} = R_w \Delta\omega_{ift} \tag{5.22}$$

$$\Delta\omega_{ift} = \dot{\omega_{ift}} \Delta t_{ft} \tag{5.23}$$

$$\dot{\omega}_{ift} = \frac{S_{ift}}{R_w \cdot \Delta t_{ft}} \tag{5.24}$$

For the second turn, the same exact process is utilized, except that the arc length of the distance traveled S_{ift} is a function of δ instead of σ.

For the linear movement, because all the wheels must travel at the same speed to move straight, assume that all wheels can rotate at the peak wheel velocity. Similar to calculations for the turns, the distance traveled can be expressed as function of the angular displacement of the wheels, as well as the angular velocity of the wheels, which combined can solve for the elapsed time of the linear movement.

$$\Delta\omega_m = \frac{L}{R_w} \tag{5.25}$$

$$\Delta\omega_m = \dot{\omega}_m \cdot \Delta t_m \tag{5.26}$$

$$\Delta t_m = \frac{L}{R_w \cdot \Delta\omega_m} \tag{5.27}$$

6 Inverse Kinematics

Arm

Inverse Kinematics can be thought of as the opposite of Forward Kinematics, such that it solves for all of the possible sets of joint angles that resolves the end effector to a desired position relative to the robot's base frame. In this case, H is the desired position and orientation. Equation 6.1 is the equation where one or more solutions needs to be solved in order to find the joint angles (q_i).

$$T_n^0(q_1, ..., q_n) = H_1(q_1) \cdots H_n(q_n) = H \tag{6.1}$$

Inverse kinematics can be far more complex than solving forward kinematics. In some cases an unsolvable problem will be encountered. A typical approach to solving inverse kinematics problems is called **kinematic decoupling**. Essentially it approaches the problem by breaking it down into two subproblems, with the first is calculating the position of the wrist center (intersection between the wrist axes) and then finding the orientation of the wrist center. [2]

As mentioned in the forward kinematics section, x_n^0 and y_n^0 can be thought of as a projection of the length of the arm onto the (x_0,y_0) plane. This is reaffirmed when solving for the position of the base joint, $\theta 1$, because this ratio is all that is needed to find its only solution, as seen in Equation 6.2.

$$\begin{aligned} \frac{y_n^0}{x_n^0} &= \frac{s_1[d_2 s_2 + a_2 c_2 + d_3 s_{23} + (d_4 + d_5) s_{234}]}{c_1[d_2 s_2 + a_2 c_2 + d_3 s_{23} + (d_4 + d_5) s_{234}]} \\ &= tan(\theta_1) \end{aligned} \tag{6.2}$$

Taking the inverse tangent of both sides of this equation results in the solution for θ_1, as seen in Equation 6.3.

$$\theta_1 = atan2(y_n^0, x_n^0) \tag{6.3}$$

θ_1 is the only revolution about the base's z-axis. Therefore, the subsection of the arm following θ_1 can be thought of as a planar manipulator in the (x_2, y_2) plane, meaning all joint rotations act in parallel to the z_2 axis. For algebraic convenience, the "zero position" of this planar manipulator should point in the positive x direction of the plane it moves in. Therefore, a convenience frame i is defined as one that is offset from frame 2 by a rotation of $-\frac{\pi}{2}$ about its z-axis. Equations 6.4 and 6.5 resolve the end-effector's x and y position into this frame i.

$$x_n^i = d_2 c_2 + a_2 s_2 + d_3 c_{23} + (d_4 + d_5) c_{234} \tag{6.4}$$

$$y_n^i = d_2 s_2 + a_2 c_2 + d_3 s_{23} + (d_4 + d_5) s_{234} \tag{6.5}$$

Because this subsection acts as a planar manipulator in the (x_i, y_i) plane, Equation 6.6 holds as such:

$$z_n^i = 0 \tag{6.6}$$

Perform the frame transformations seen in Equation 6.7 to resolve the position of the end-effector in the i^{th} frame and subsequently calculate x_n^i and y_n^i.

$$P_n^i = T_0^i \cdot P_n^0 = (T_1^0 \cdot T_2^1 \cdot T_i^2)^{-1} \cdot P_n^0 \tag{6.7}$$

Where:

$$P_n^i = [x_n^i \; y_n^i \; z_n^i]^T, \; P_n^0 = [x_n^0 \; y_n^0 \; z_n^0]^T \tag{6.8}$$

θ_4 can be eliminated from Equations 6.9 and 6.10 by substituting in Equation 4.13.

$$x_n^i = d_2 c_2 + a_2 s_2 + d_3 c_{23} + (d_4 + d_5) c_\gamma \tag{6.9}$$

$$y_n^i = d_2 s_2 + a_2 c_2 + d_3 s_{23} + (d_4 + d_5) s_\gamma \tag{6.10}$$

Furthermore, define the variables x' and y' as the difference between x_n^i and y_n^i and the length of the linkage distal to the wrist joint in the x and y axes, respectively, as seen in Equations 6.11 and 6.12. Substituting them into Equations 6.9 and 6.10 yields Equations 6.13 and 6.14

$$x' = x_n^i - (d_4 + d_5) c_\gamma \tag{6.11}$$

$$y' = y_n^i - (d_4 + d_5) s_\gamma \tag{6.12}$$

$$x' - d_2 c_2 + a_2 s_2 = d_3 c_{23} \tag{6.13}$$

$$y' - d_2 s_2 - a_2 c_2 = d_3 s_{23} \tag{6.14}$$

From here, squaring both sides of both equations, adding them together, and simplifying by grouping the c_2, s_2, and the rest of the terms together, results in Equation 6.15.

$$(-2x' d_2 - 2y' a_2) c_2 + (2x' a_2 - 2y' d_2) s_2 + (x'^2 + y'^2 + d_2^2 + a_2^2 - d_3^2) = 0 \tag{6.15}$$

This follows the general form of Equation 6.16. To solve this general equation for its angle β, define the angle δ with Equation 6.17.

$$P c_\beta + Q s_\beta + R = 0 \tag{6.16}$$

$$c_\delta = \frac{P}{\sqrt{P^2+Q^2}},\ s_\delta = \frac{Q}{\sqrt{P^2+Q^2}} \tag{6.17}$$

Therefore:

$$\delta = atan2\left(\frac{Q}{\sqrt{P^2+Q^2}},\ \frac{P}{\sqrt{P^2+Q^2}}\right) \tag{6.18}$$

$$c_\delta c_\beta + s_\delta s_\beta + \frac{R}{\sqrt{P^2+Q^2}} = 0 \tag{6.19}$$

Finally, for $\beta = \theta_2$:

$$\theta_2 = \delta \pm cos^{-1}\left(\frac{-R}{\sqrt{P^2+Q^2}}\right) \tag{6.20}$$

Where:

$$P = -2x'd_2 - 2y'a_2 \tag{6.21}$$

$$Q = 2x'a_2 - 2y'd_2 \tag{6.22}$$

$$R = x'^2 + y'^2 + d_2^2 + a_2^2 - d_3^2 \tag{6.23}$$

Solving for c_{23} and s_{23} from Equations 6.11 and 6.12 yields Equations 6.24 and 6.25. Dividing them by one another results in Equation 6.26, and taking the inverse tangent of both sides allows for the solution seen in Equation 6.27.

$$c_{23} = \frac{x' - d_2c_2 + a_2s_2}{d3} \tag{6.24}$$

$$s_{23} = \frac{y' - d_2s_2 - a_2c_2}{d3} \tag{6.25}$$

$$tan(\theta_2+\theta_3) = \frac{y' - d_2s_2 - a_2c_2}{x' - d_2c_2 + a_2s_2} \tag{6.26}$$

$$\theta_3 = atan2(y' - d_2s_2 - a_2c_2, x' - d_2c_2 + a_2s_2) - \theta_2 \tag{6.27}$$

Finally, θ_4 can be solved using Equation 4.13.

$$\theta_4 = \gamma - \theta_2 - \theta_3 \tag{6.28}$$

Therefore, θ_1 has up to a single unique solution, θ_2 has up to two unique solutions, and both θ_3 and θ_4 have up to a single unique solution for each unique solution of θ_2. In other words, there are either zero, one, or two unique solutions to this overall inverse kinematic problem.

Combined

With the combination of vehicle velocity kinematics and the arm's inverse kinematics, the vehicle and subsequent arm movement can be calculated such that the vehicle can traverse the surface to approach a sample at a known (x,y) position with a desired orientation θ relative to it, as well as a the desired orientation of the robot's gripper relative to the surface γ.

7 Scope of Achievement

For the scope of achievement and study, milestones and careful planning had to been made. Most of the concepts learned in this class were able to translate over to this project. A list of milestones are presented below, in order of planning and achievement.

1. Build the vehicle part of the rover
2. Create the ROS workspace environment
3. Create a terrain similar to Mars terrain in Gazebo
4. Test the vehicle URDF and tune the controls
5. Build the arm for the rover
6. Calculate and create scripts for the kinematics of the arm

The CAD model was successfully created in SolidWorks for both the arm and the vehicle. Small adjustments had to be made regarding the steering joints since there was no good reference for it. The arm model matched the rovers well. The URDF had some issues exporting properly for both the rover and the arm because sometimes axes werent defined properly but they work correctly in Gazebo. The models for both of the arm and vehicle render in and, with tele-op successfully controlling joints. The controls were tuned so that the rover did not fly around in the world nor did the wheels skid around. C++ was used to write the scripts for all the kinematics. They calculate the forward, inverse and velocity kinematics. DH frames and the DH table were calculated manually.

8 Model Validation and Testing

The models, mechanical and computational, were extensively tested in Gazebo. There were a total of three different environments, one for just the vehicle, one for just the arm (mounted to the world), and one with both, where the arm is mounted to the lower portion of the chassis, as seen in Figure 8.1.

As mentioned before, the models were created in SolidWorks, which is how the render can occur. Furthermore, the kinematic equations were implemented in C++. ROS was utilized to connect the Gazebo joint controllers to the outputs of the various kinematic calculation servers. These files of interest are located in the SRR workspace directory as follows:

- Arm Forward and Inverse Kinematics Server: ArmKinContainer and ArmKinCalculator in the SRR's kinematics package
- Vehicle Velocity Kinematics Server: VehicleVelKinContainer and VehicleVelKinCalculator in the SRR's kinematics package
- Combined Inverse Kinematics Server: CombinedKinContainer and CombinedKinCalculator in the SRR's kinematics package

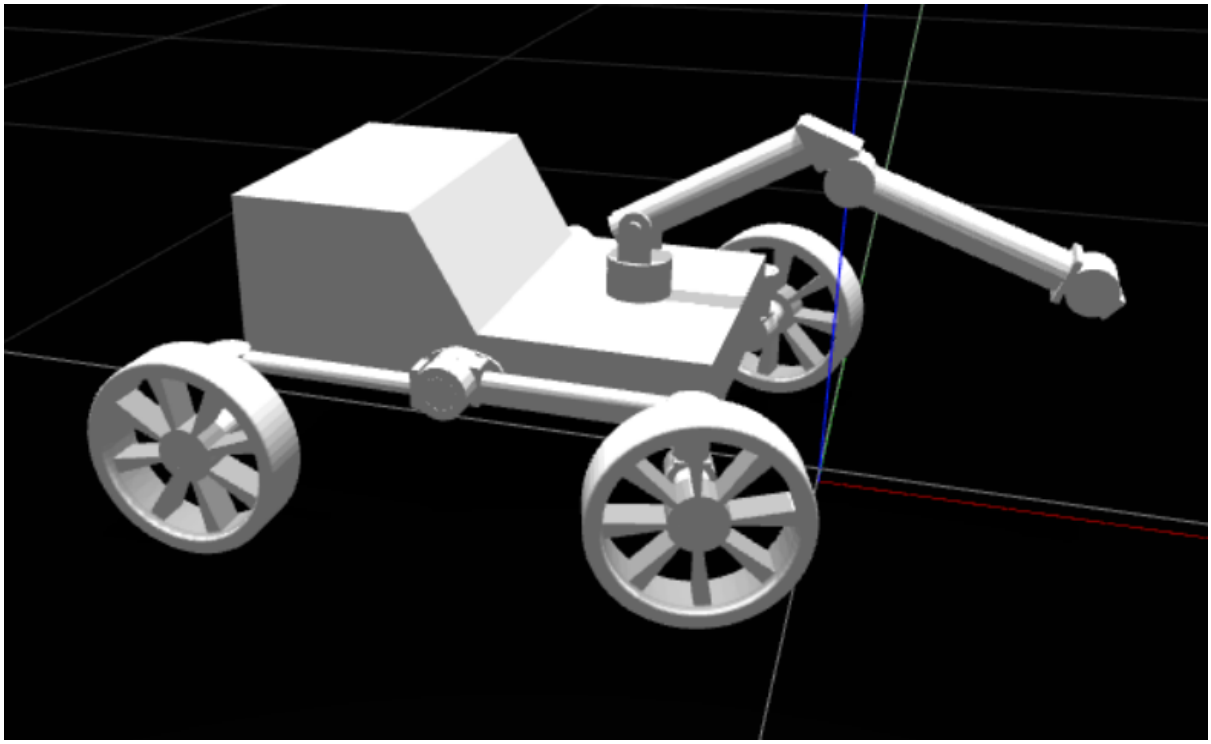

Figure 8.1: Gazebo Simulation of the Combined Vehicle and Arm Models

9 Conclusion

Designing a cad model without much reference was quite difficult to get the dimensions of the robot to look right, some parts had to be redesigned. The fallback goal is to have the robot and arm operating in separate gazebo environments so that the controls algorithm can be tested independently, and this was successfully implemented. A lot of the concepts learnt from class were used for this project. Knowing how to do kinematics of a robot was very beneficial for calculating kinematics of the arm. Even though that the arm is four degrees of freedom, the arm would not properly turn in Gazebo because it needed dummy frames. As a result, writing out and assigning dummy frames fixed the issue. The robot was successfully created in SolidWorks and exported to a URDF. The controls were tuned accordingly in Gazebo so that the rover can be operated via tele-op. Calculations were determined and scripted. Overall a lot was learned from this project and it was a good experience.

Bibliography

[1] John J. Craig. *Introduction to Robotics: Mechanics and Control*. 2nd ed. Reading, Massachusetts: Addison-Wesley, 1989. ISBN: 0201095289.

[2] S. Hutchinson M. W. Spong and M. Vidyasagar. *Robot Modeling and Control*. 1st ed. John Wiley and Sons, Inc., 2001.

[3] T. L. Huntsberger P. S. Schenker and G. T. McKee University of Reading (UK) P. Pirjanian Jet Propulsion Laboratory (USA). "Robotic Autonomy for Space: Cooperative and Reconfigurable Mobile Surface Systems". In: i-SAIRAS 2001. Canadian Space Agency, St. Hubert, Quebec, Canada, 2001.