



Open Source Rover: Software Controls

Author: Eric Junkins, Michael Cox



Jet Propulsion Laboratory
California Institute of Technology

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology. ©2018 California Institute of Technology. Government sponsorship acknowledged.

Contents

1 Control System	3
1.1 Drive Motor calculations	3
1.2 Corner Motor Positions	10

1 Control System

There are two main issues we need to address when designing the control mechanics for the rover. One is the direction that each of the corner wheels needs to point in order for the rover to be able to turn correctly. The other is how fast each of the individual drive motors needs to spin (it's not the same speed at each wheel when the rover is turning!).

1.1 Drive Motor calculations

The 6 wheel rover design employs Ackerman steering, which can be seen in Figure 1. The rover steers about a point which lies on the line that passes through the two center wheels. All 6 wheels need to align their center axis towards the point around which the rover will turn. Notice that we observe a more drastic steering angle for the wheels closer to that point than for the wheels that are further away.

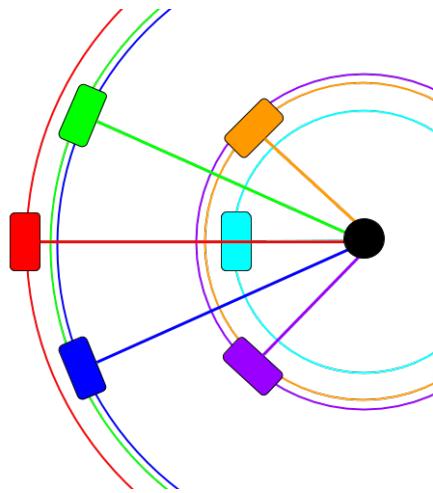


Figure 1: Ackerman Steering

Let's go step-by-step though how to calculate the speed of each of the drive wheels. To start, we will look at the rover which will be turning about a point P at radius r, shown in Figure 2

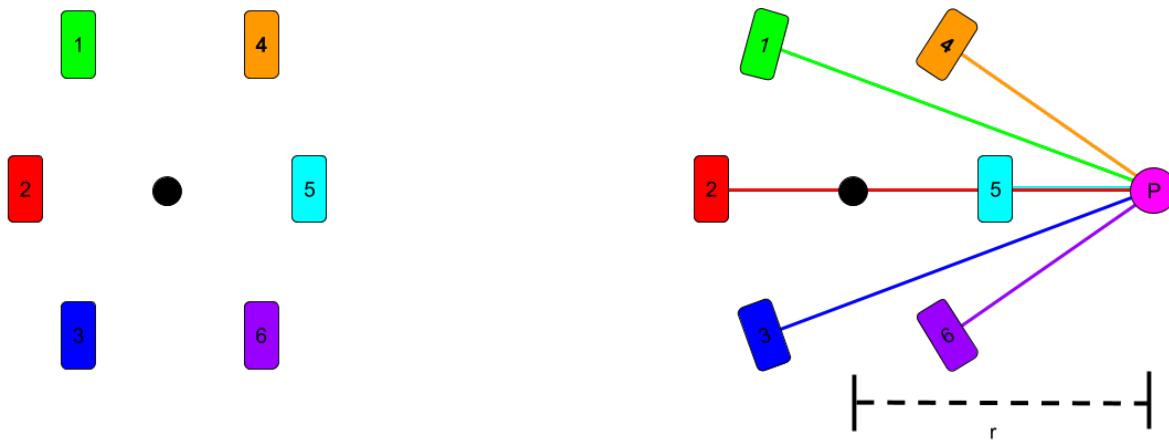


Figure 2: Rover turning about point P

We start by sending a signal from the controller for the rover to drive forward. Let's call the speed at which it drives X (for now, we set X to its maximum value of 100). By convention, this will tell the middle of the rover that it should move forwards at the max speed of 100. However, if the rover is turning, each wheel will need to spin at a slightly different rate to avoid slipping or "scrubbing". Because no wheel can spin faster than speed 100 and because the wheels have differing distances they must travel along their arcs during the turn, we send the max speed of 100 to the wheel furthest away from the point we are turning around. In the case above, this is wheel #2. Therefore, we now set wheel #2 to spin at speed 100 as shown in Figure 3.

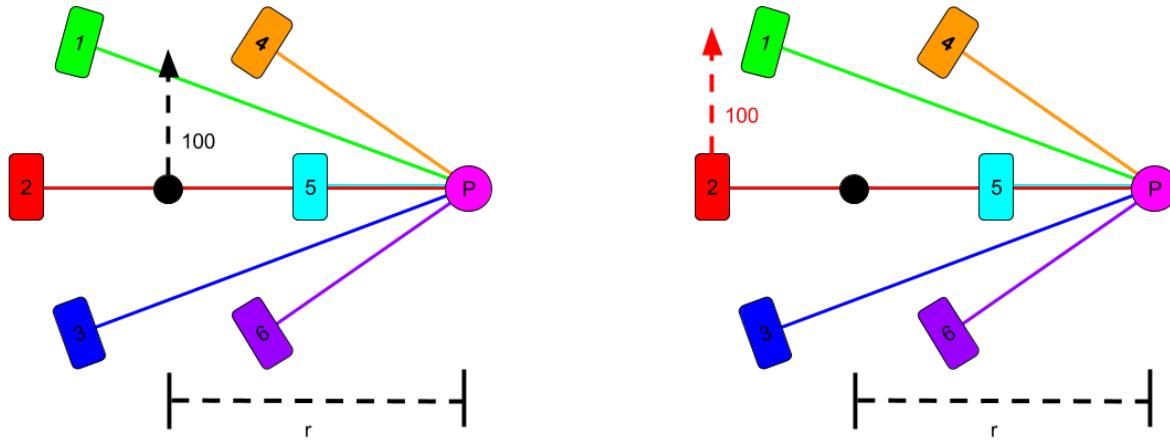


Figure 3: Rover distributing speed to fastest/furthest wheel

Next, we examine how to distribute speed to the other wheels. We will make use of arclengths of the circles the wheels turn about which is given by the relation in Equation 1:

$$S = R\theta \quad (1)$$

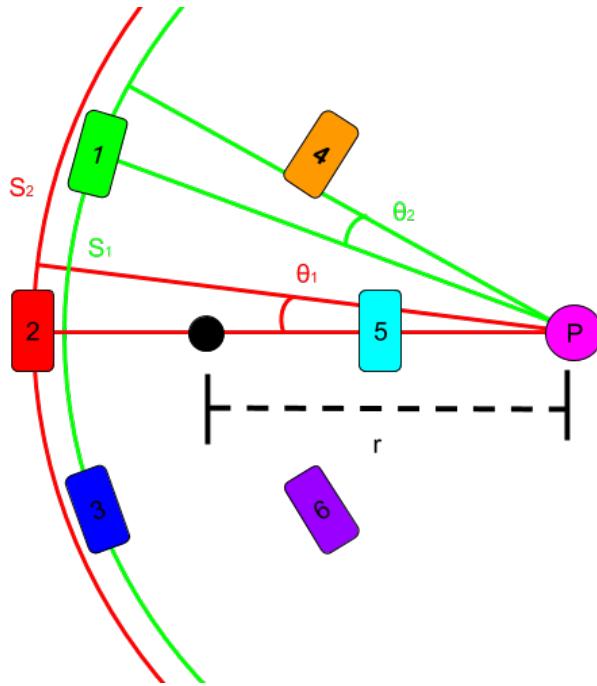


Figure 4: Arclengths of driving curves

In order for the entire rover to drive cohesively each wheel must travel the same number of degrees from Equation 1 in the same time, so we can solve for the relation between each of the wheels now. (In the equations below, θ is the angle that the wheel traverses, S is the arclength that the wheel travels, and R is the radius or distance from the wheel to the turning point)

$$\theta_1 = \frac{S_1}{R_1} \quad \theta_2 = \frac{S_2}{R_2} \quad (2)$$

$$\theta_1 = \theta_2 \quad S_1 = S_2 \frac{R_1}{R_2} \quad (3)$$

From Equation 3 we can see that the ratio of the distance traveled (and thus the speed) of each of the wheels is going to be given by the ratio of the radius of the arc that each wheel is turning about. The last step is to figure out the radius of that arc for each wheel.

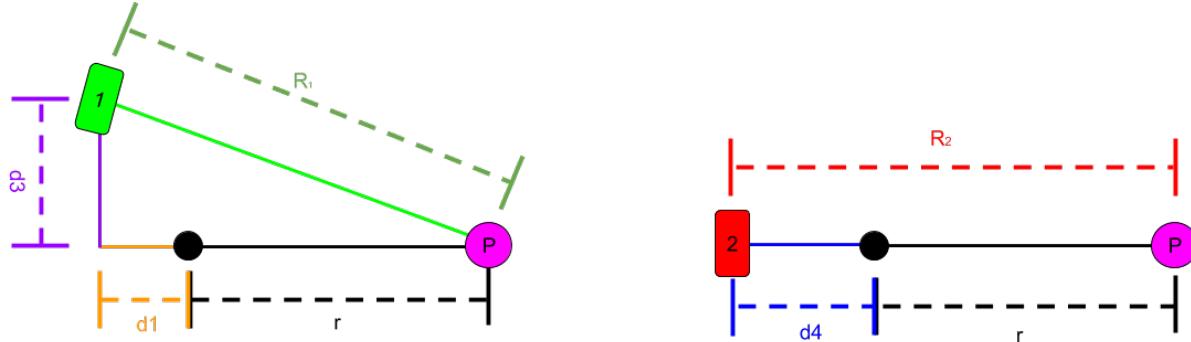


Figure 5: Radius of arc for turning

$$R_1 = \sqrt{d_3^2 + (\textcolor{brown}{d}_1 + r)^2} \quad R_2 = \textcolor{blue}{d}_4 + r \quad (4)$$

The distances d_1 , and d_3 are geometric relations based on the mechanical lengths of the

robot itself. Putting this all together, let's look at an example of calculating speed and turning distance.

Variable	Physical description
d_1	Horizontal distance between middle of rover and the corner wheels
d_2	Vertical distance between the middle of rover and back corner wheels
d_3	Vertical distance between the middle of the rover and the front corner wheels
d_4	Horizontal distance between the middle of the rover and the center wheels

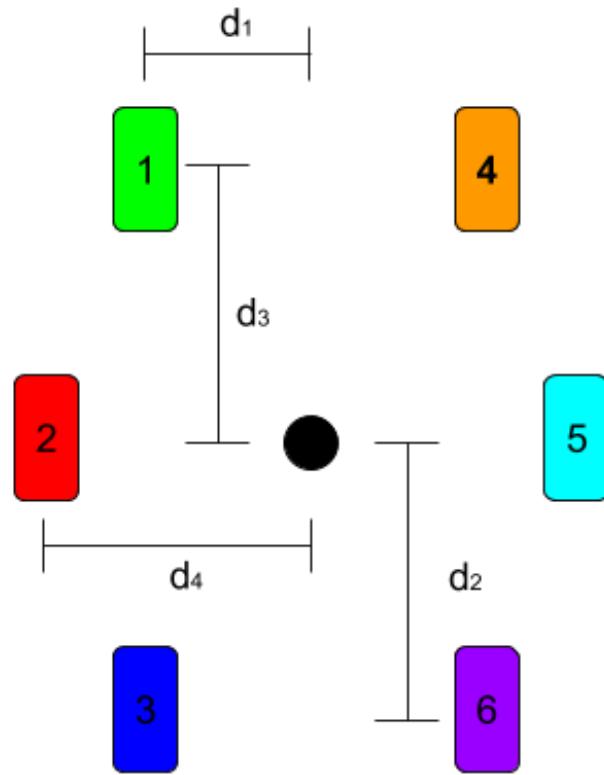


Figure 6: Physical distance of wheel geometry

For the version of the rover described in the build documents, the distances are as follows¹:

$$d_1 = 7.254\text{in} \quad d_2 = d_3 = 10.5\text{in} \quad d_4 = 10.073\text{in} \quad (5)$$

For this example, we will have a controller signal of max speed (100), and a turning radius of 30 inches (by convention the positive turning radius is turning to the right and negative is turning to the left), as shown in Figure 7:

$$r = 30\text{in} \quad X = 100 \quad (6)$$

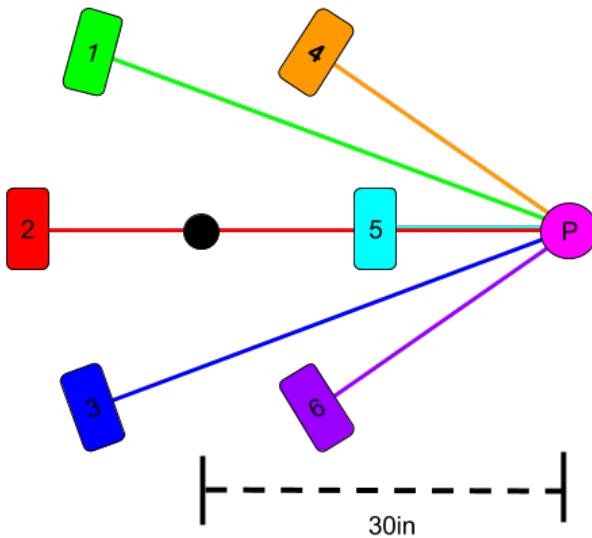


Figure 7: 30 Inch Turning radius

Using Equations 3 and 4 we get the following relations for the speed distribution. We substitute Equation 4 into 3, which solves for the arc length distance. This is analogous to the speed of each wheel, as speed is just $\text{speed} = \frac{\text{distance}}{\text{time}}$. We denote speed with V (for velocity), and we now have:

¹If you change the rover's geometry (such as distances between wheels) you will need to update these values in the code as well!

$$V_1 = X \frac{R_1}{R_2} = X \frac{\sqrt{d_3^2 + (d_1 + r)^2}}{r + d_4} \quad (7)$$

$$V_2 = X \quad (8)$$

$$V_3 = X \frac{R_3}{R_2} = X \frac{\sqrt{d_2^2 + (d_1 + r)^2}}{r + d_4} \quad (9)$$

$$V_4 = X \frac{R_4}{R_2} = X \frac{\sqrt{d_3^2 + (r - d_1)^2}}{r + d_4} \quad (10)$$

$$V_5 = X \frac{R_5}{R_2} = X \frac{r - d_4}{r + d_4} \quad (11)$$

$$V_6 = X \frac{R_6}{R_2} = X \frac{\sqrt{d_2^2 + (r - d_1)^2}}{r + d_4} \quad (12)$$

Substituting in the values for the speed X and the various distances, we obtain the following values for the motor speeds:

Motor	Motor speed % of max (100)
1	96.59
2	100
3	96.59
4	62.52
5	49.73
6	62.52

We can see that at a small turning radius like 30 inches, we will see a drastic difference between the motor speeds. As the turning radius increases, we will see less and less difference between the speeds. This is the case up until the point where the calculated turning radius is

250 inches or greater; 250 inches is the point at which the encoders on the corners no longer have the resolution to differentiate between 1° and 0° and thus we say that anything above a turning radius of 250 inches is not turning at all and we give all motors the same speed.

1.2 Corner Motor Positions

To signal turning, the rover will take values from the controller from -100 to 100. This value represents the percentage of the minimum (tightest) turning radius that the wheels can handle, with negative numbers turning to the left and positive numbers to the right. The higher the magnitude of the number, the tighter turning radius, i.e. -100 means the tightest turn possible to the left, 0 means straight ahead, and 100 means the tightest turn possible to the right. This "tightest" turning radius is determined by a combination of the geometric distances of the rover as well as the physical hard stop limitations of the corner motors. In our system, each of the corner motor is allowed to turn up to 45 degrees before hitting a physical hard stop, as shown in Figure 8.

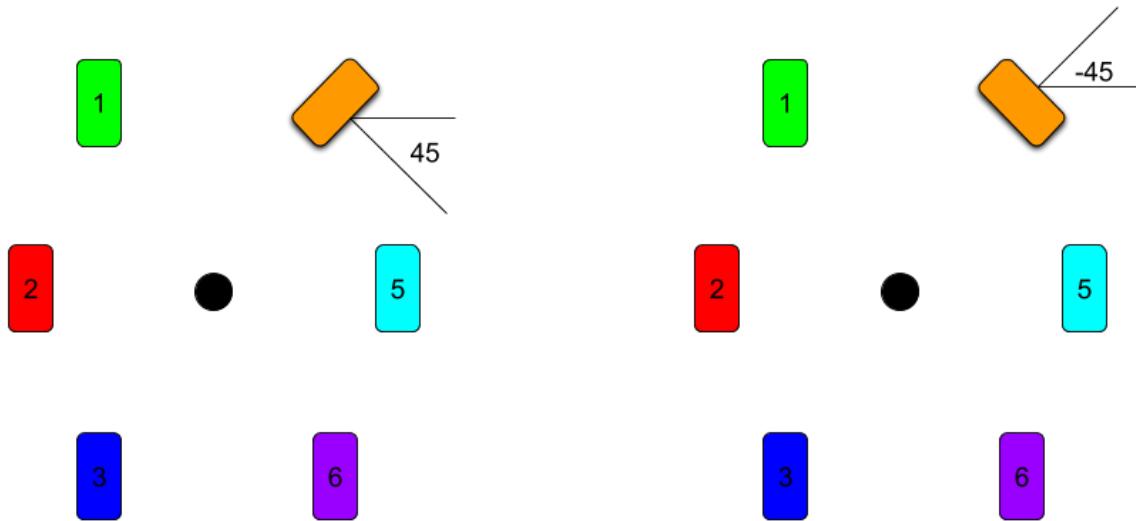


Figure 8: Corner angle limitations

In order to figure out the tightest turning radius, we need to solve for the closest place along the rover's center axis that the corner wheel can point given its 45° limitation. This is shown in Figure 9:

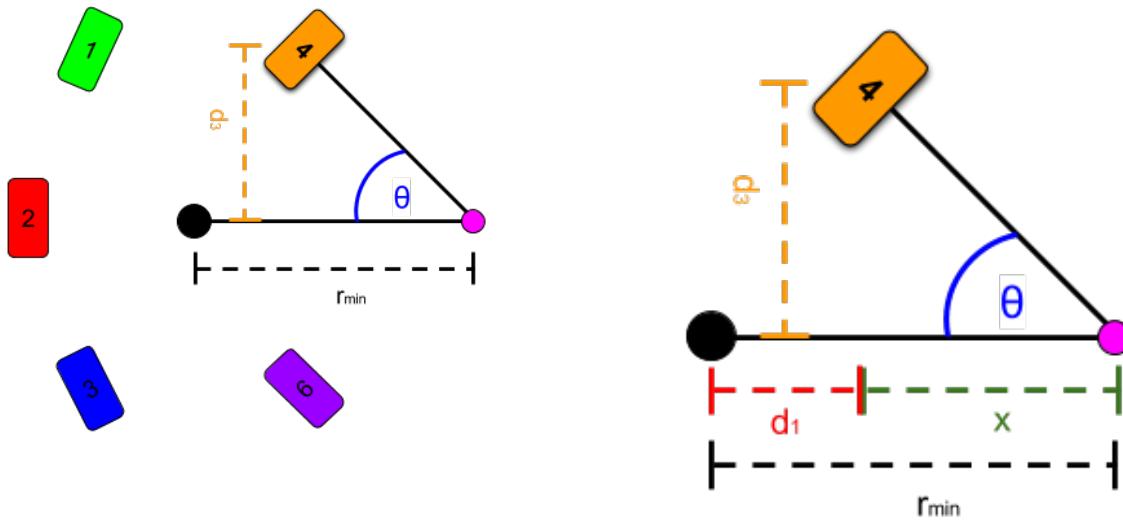


Figure 9: Minimum Turning Radius calculations

$$\tan(\theta) = \frac{d_3}{x} \quad r_{min} = d_1 + x \quad (13)$$

$$r_{min} = d_1 + \frac{d_3}{\tan(\theta)} \quad r_{min} = 7.254 + \frac{10.5}{\tan(45^\circ)} \quad (14)$$

$$r_{min} = 17.754 \text{ inches} \quad (15)$$

Plugging in the distance values from our physical robot, we see that the minimum turning radius to each side will be at 17.754 inches. In order to not push the system to the physical hard stops when it turns, we therefore set the minimum radius to be at 20 inches to either side. As mentioned before, the "maximum" turning radius is at 250 inches to either side, as this is the location in which the encoder no longer can differentiate between it being at 0°

and 1° . In order to calculate what target angle each corner should be at given the turning radius input from the controller, we take Equation 13 and solve for θ :

$$\tan(\theta) = \frac{d_3}{x} \quad \theta = \tan^{-1}\left(\frac{d_3}{R - d_1}\right) \quad (16)$$

Where R is the physical turning radius of the rover (but remember that the values from the controller are given as a percentage of the range of the possible turning radii for the rover).