



# Open Source Rover: Software Controls

Author: Eric Junkins



**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

<b>1 Control System</b>	<b>3</b>
1.1 Drive Motor calculations . . . . .	3
1.2 Corner Motor Positions . . . . .	10

# 1 Control System

There are two systems we have to look at when designing the controls mechanics for the rover. One is how fast each of the drive motors must move, and the other is where each of the corners need to point in order for the rover to be able to make turns.

## 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 as they are both fixed. All 6 wheels are lie perpendicular to the vector from that point to the corner wheel, which we can see causes a more drastic angle in steering for the wheels closer to the point than further away.

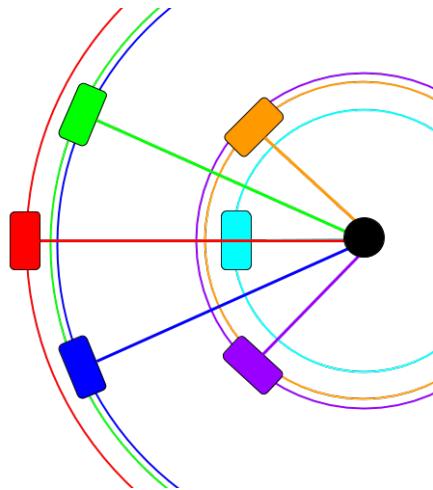


Figure 1: Ackerman Steering

Lets 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, shows in Figure 2

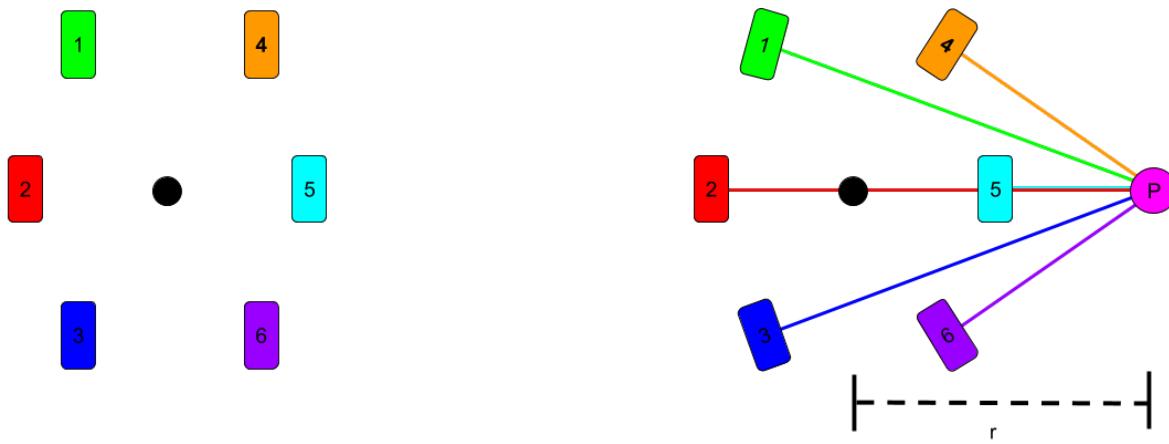


Figure 2: Rover turning about point p

From the controller we will send a signal for the rover to drive forward. Lets call that speed X, and assume we sent a speed of 100. By convention this will tell the middle of the rover it should move at the max speed of 100. If however we are turning the wheels will need to distribute that speed differently to each wheel. In order for the middle of the rover to move at it's max speed we will distribute the max speed to the furthest out wheel, in this case that is wheel #2 (it will always be the middle wheel furthest from the turning point). So we now choose that the speed 100 is distributed to wheel 2 shows in Figure 3.

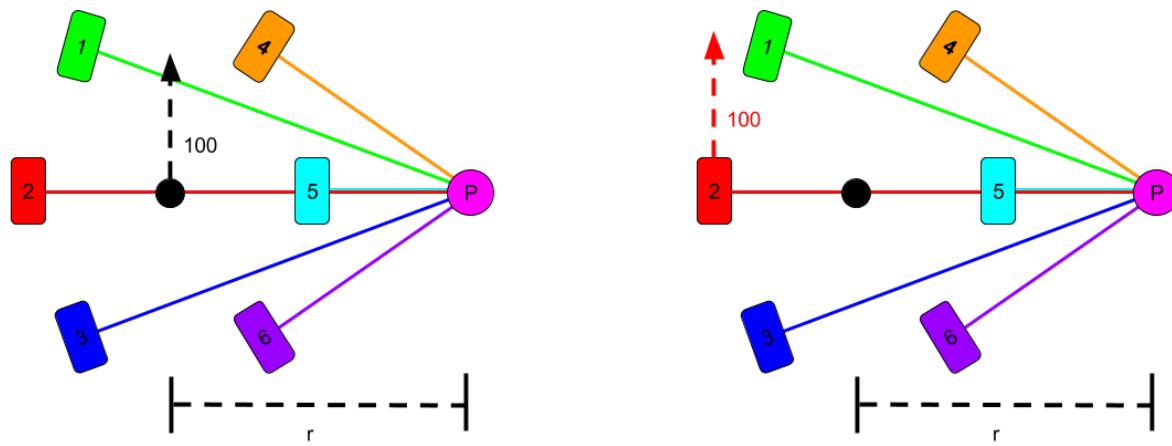


Figure 3: Rover distributing speed to fastest wheel

Now we look at how to distribute speed to the other wheels. We will make use of arclengths of the circles the wheels turn about, and the relation in Equation 1

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

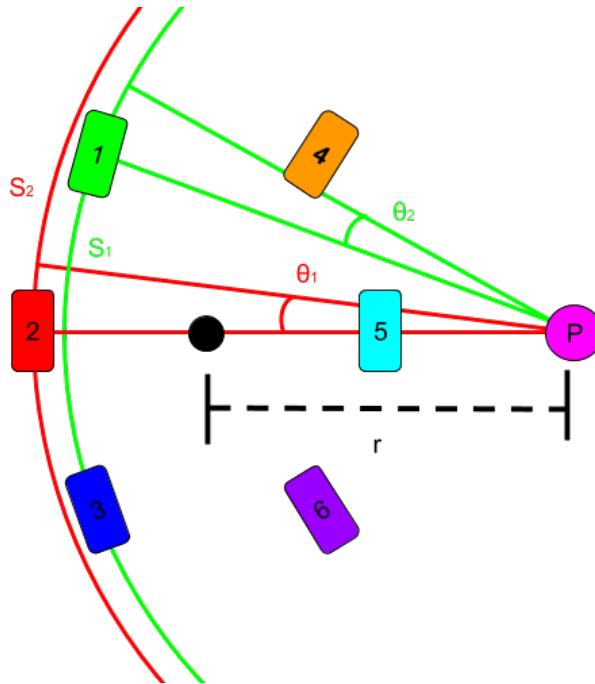


Figure 4: Arclengths of driving curves

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.

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

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

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

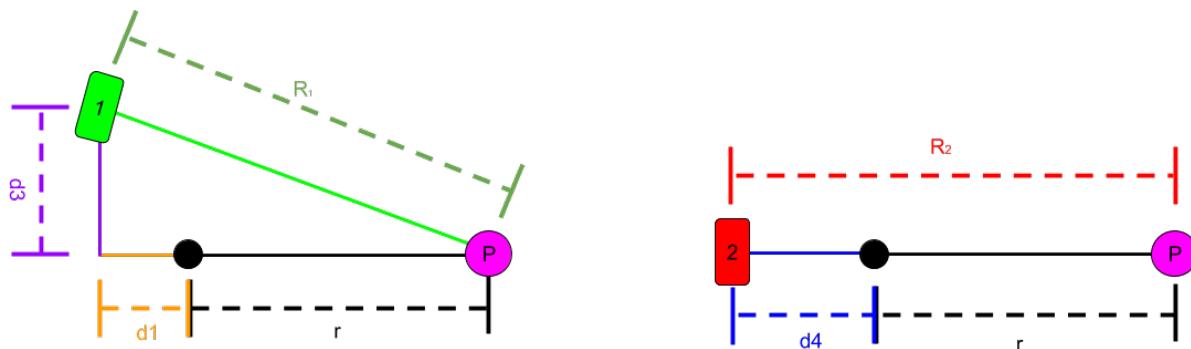


Figure 5: Radius of arc for turning

$$R_1 = \sqrt{d_3^2 + (d_1 + r)^2} \quad R_2 = d_4 + r \quad (4)$$

These distances  $d_1$ , and  $d_3$  are geometric relations based on the mechanical lengths of the robot itself. Putting this all together lets look at an example speed and turning distance.

Variable	Physical description
$d_1$	Horizontal distance between middle of rover and the corner wheels
$d_2$	Verticle distance between the middle of rover and back corner wheels
$d_3$	Verticle 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

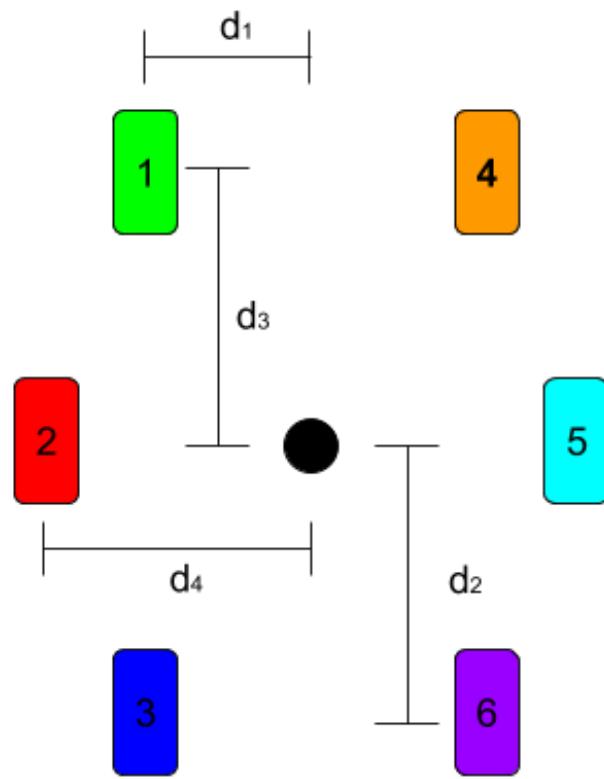


Figure 6: Physical distance of wheel geometry

For the base model of the Rover the distances are as follows <sup>1</sup>:

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

<sup>1</sup>If you change aspects such as distances between wheels in the rover you will need to update these in the code as well

For this example we will have a controller signal of max speed, being 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), depicted by Figure 7

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

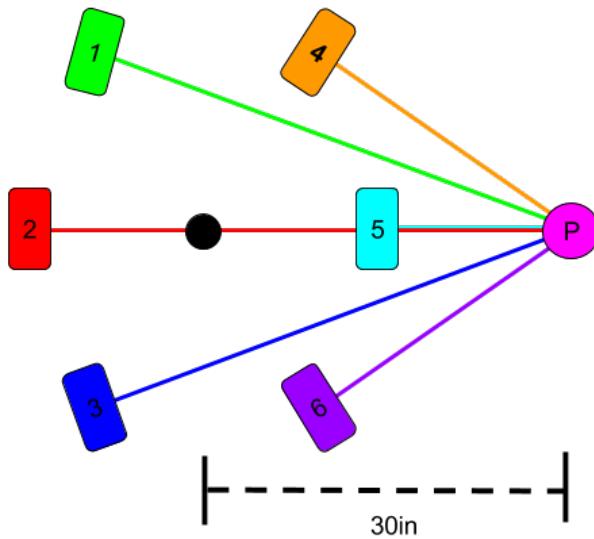


Figure 7: 30 Inch Turning radius

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 the  $speed = \frac{distance}{time}$ .

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

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

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

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

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

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

Substituting in the values of the speed X and distances it leads to the motors spinning at the following speed % (out of 100).

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 such as 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, up until the point where the calculated turning radius is 250 inches or greater. This is the point in which the encoders on the corners no longer have spacial 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 it gives all motors the same speed.

## 1.2 Corner Motor Positions

The rover will take input from the controller which will be in the range of -100 to 100 inclusive. This is the percent of turning radius the wheels should be at, with negative numbers turning to the left, positive numbers to the right, and the higher the magnitude the number the tighter turning radius, ie. -100 means the tightest turn possible to the left side. 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. Each of the corner motor is allowed to turn up to 45 degrees before hitting a physical hard stop, shows in Figure 8.

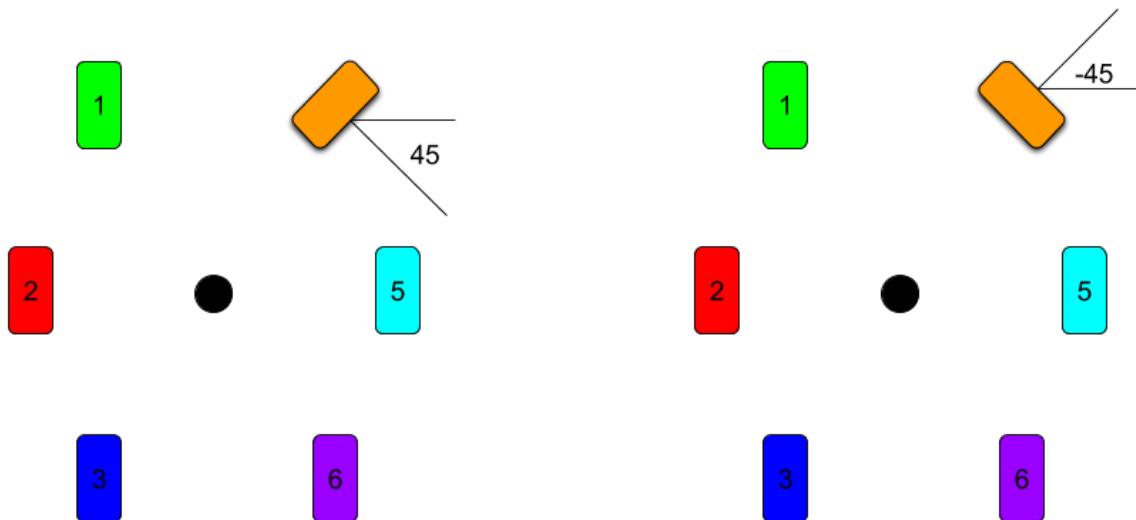


Figure 8: Corner angle limitations

In order to figure out the tightest turning radius we will need to solve for the closest point the corner wheel can be perpendicular to at its  $45^\circ$  limitation in Figure 9.

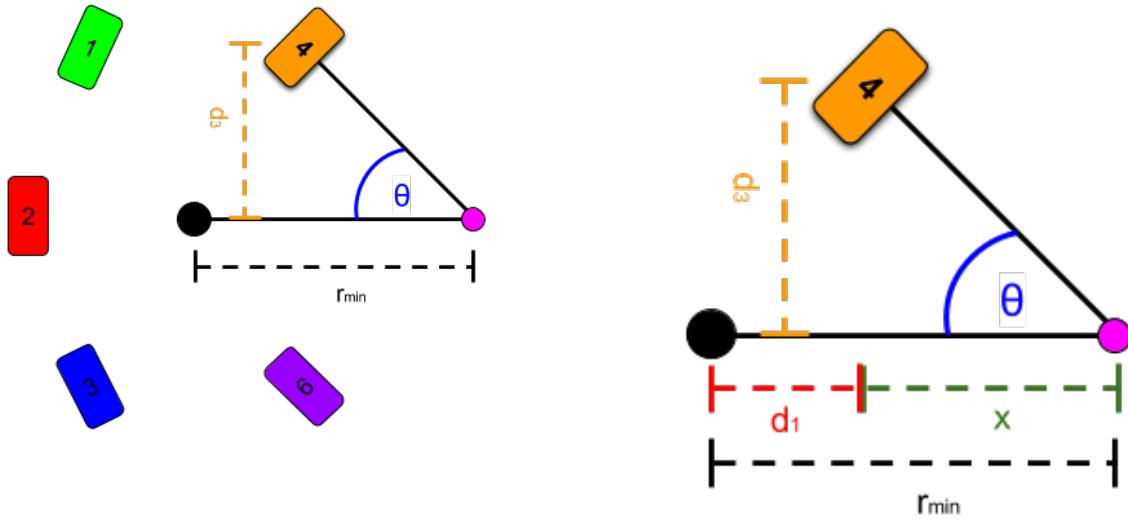


Figure 9: Minimum Turning Radius calculations

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

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

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

Here we can 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 in the software it is set to be at 20 inches to either side. As mentioned before the "minimum" 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  $0^\circ$  and  $1^\circ$ . In order to calculate what target angle each corner should be at given the turning radius input from the controller we will work backwards through Equation 13.

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

Where R is the input turning radius from the controller.