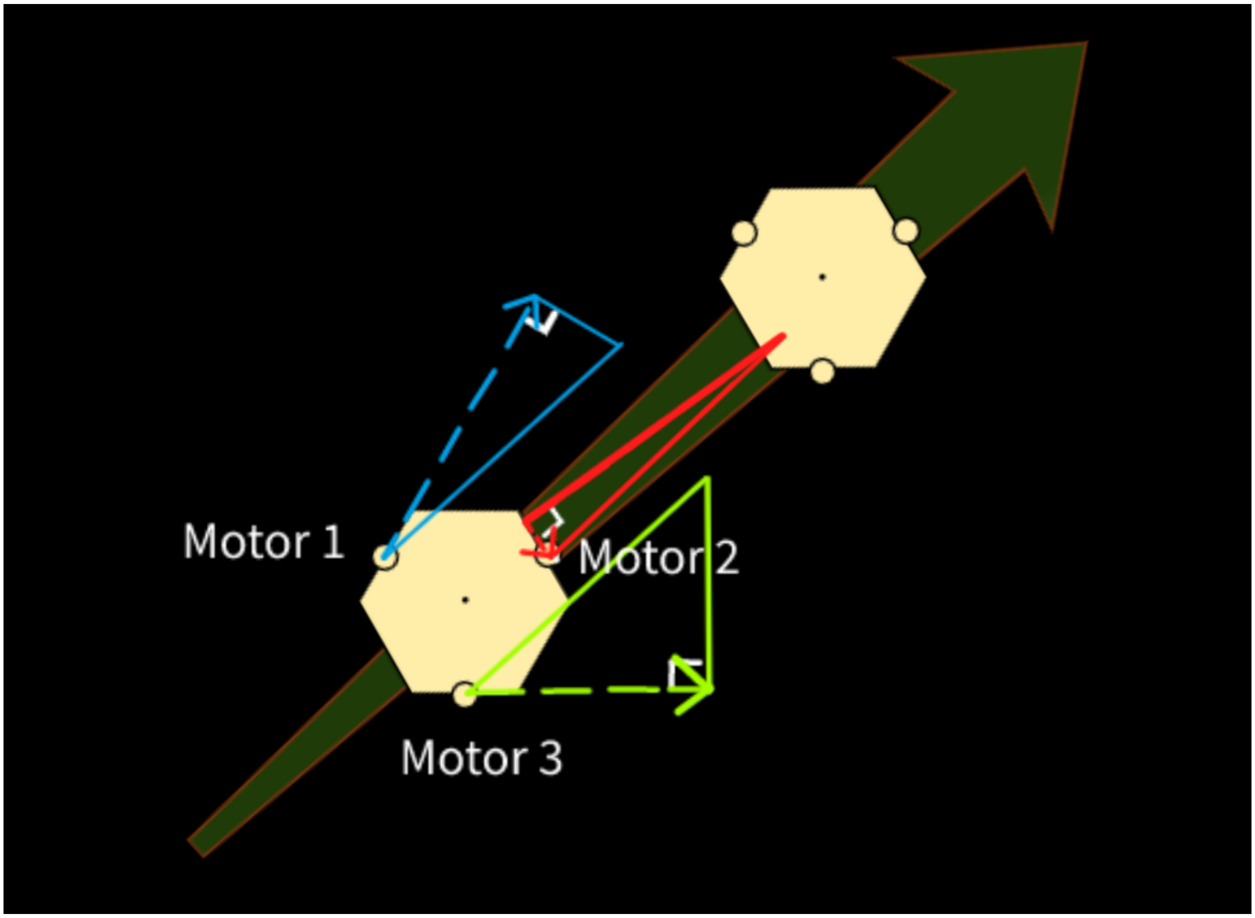
Before we begin, we must understand how a PS2 Controller joystick works. It works like a potentiometer with 2 axes (x and y). When the joystick is moved, it returns how much it moved in the x (horizontal) direction, and how much it moved in the y (vertical) direction. In short, it returns an x and y coordinate value. These coordinate values can be negative or positive depending on how the joystick is moved. This idea is utilized to derive a formula for each motor's speed.



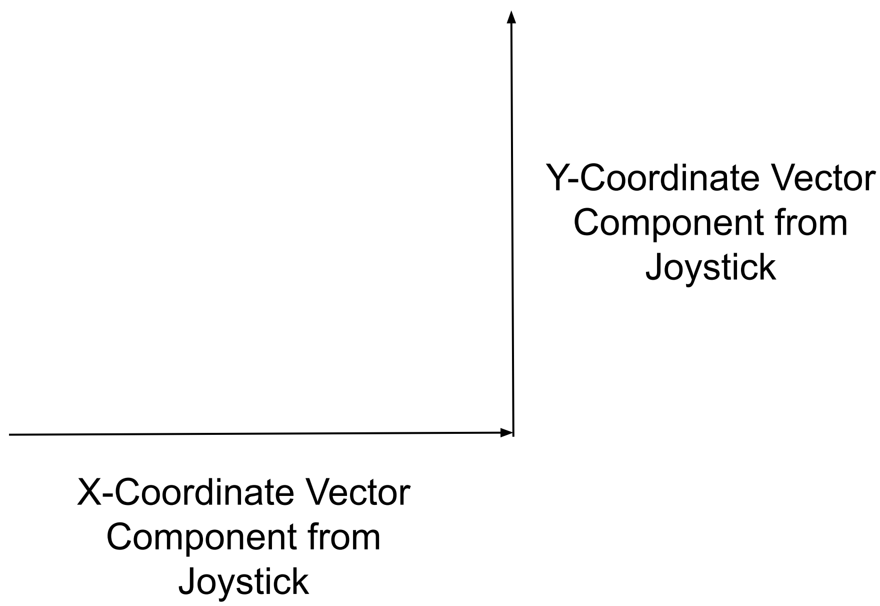
<u>Figure Above:</u> The hypotenuse (longest side) of each triangle represents the magnitude and relative direction of each corresponding motor. They all lie roughly parallel to the omni-bot's direction of travel (as indicated by the dark green arrow) and serve as a vector. Breaking each vector down into components gives us an x and y component, one of which lines up exactly with the motor's true direction (as indicated by the dotted arrow).

We know that each hypotenuse represents a vector, and assuming each vector refers to the controller's input, we can calculate the motor's speed using the equation:

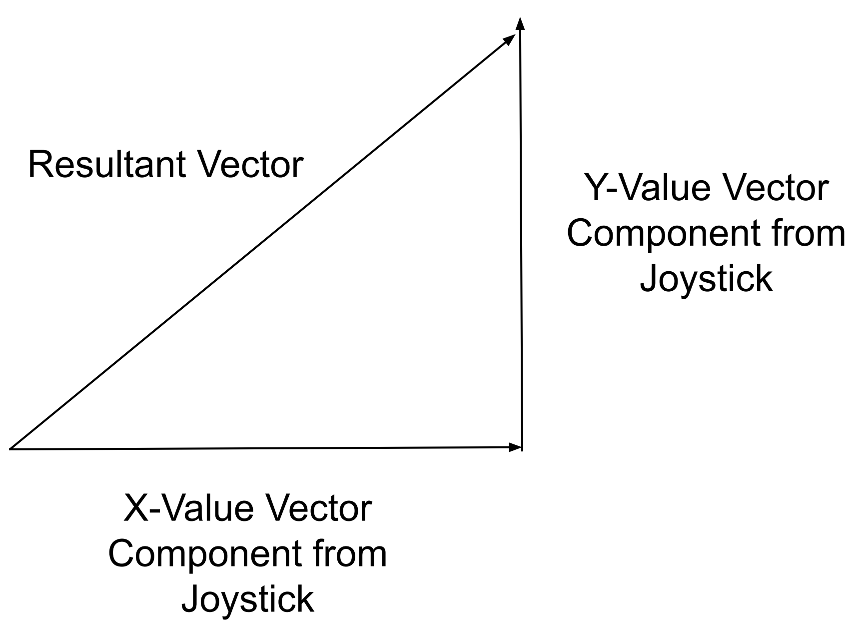
$$Vspeed = cos(theta) * controller\ input$$

Where $theta$ is the angle between the hypotenuse and dotted arrow.

However, this equation wouldn't be compatible with a PS2 controller. Remember, the joystick of a PS2 controller returns an x and y coordinate (how far it moved on both axes). Since we know that the magnitude of each coordinate directly corresponds to how far the joystick is pushed, we can also say that it is directly proportional to how *fast* the motor should be moving. Thus, we can view these x and y coordinates as vector components. If we draw the x and y vector components so that they are perpendicular to one another, we can see it forms part of a triangle.



Since all vector components form a resultant vector (hypotenuse), we can calculate the resultant vector of these two components using the Pythagorean Theorem (this resultant vector is the $controller\ input$.



This gives us the equation:

$$controller\ input\ =\ \sqrt{(Xval)^2\ +\ (Yval)^2}$$

Now how do we find $theta$? Well, let's use Motor 1 as an example:

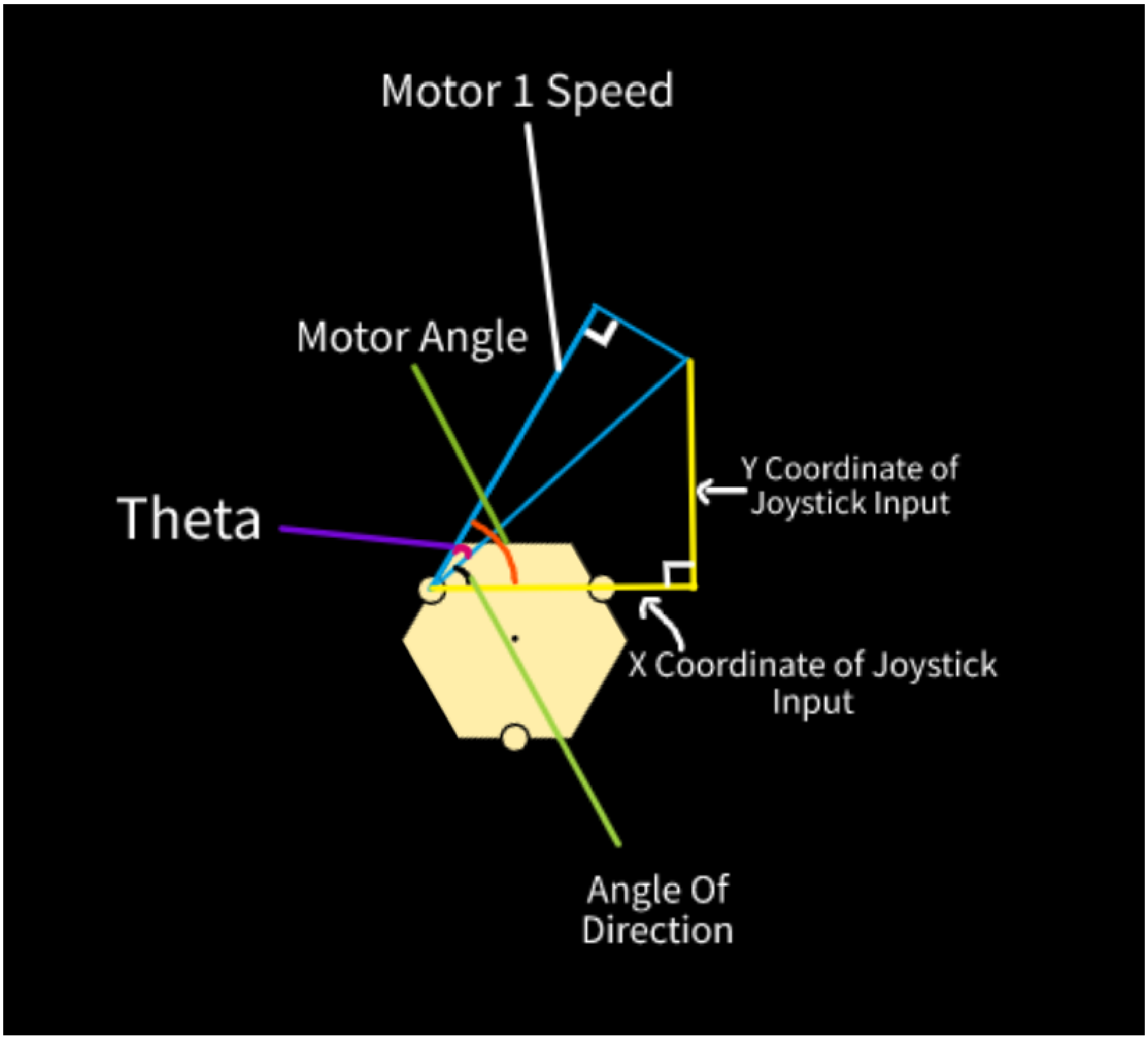


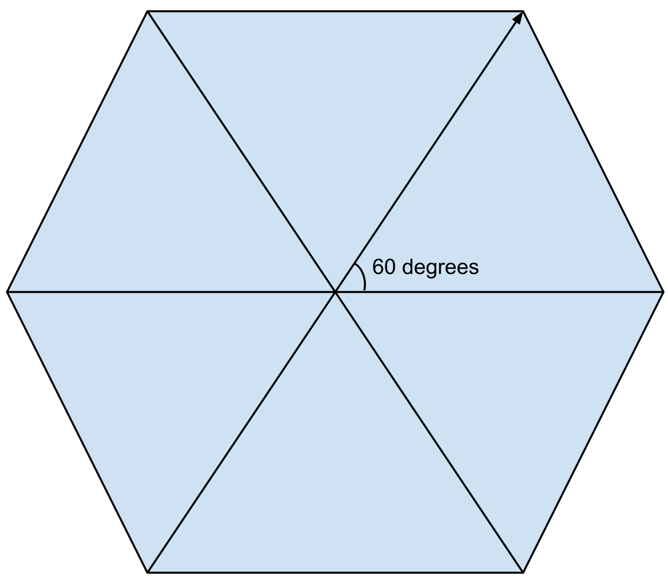
Figure Above: You will realize that there are two triangles and that both triangles share the same hypotenuse. The yellow lines represent the x and y coordinates returned from the joystick as vectors, while the blue lines form the same triangle seen in Figure A for motor 1. Using this, we can start to see how $theta$ can be calculated.

In the diagram, $theta$ is labeled as "Theta". We can see that $theta$ can be calculated using this equation:

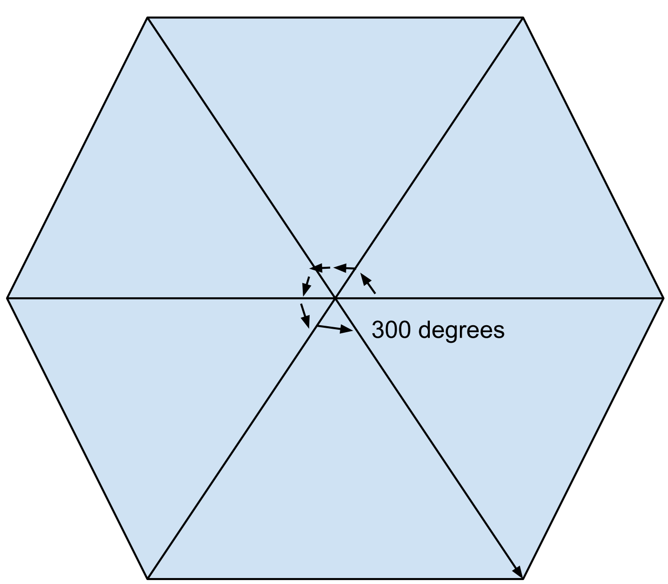
$$theta = Motor\ Angle - Angle\ of\ Direction$$

However, to get $theta$, we must figure out what the "Motor Angle" and the "Angle of Direction" equals.

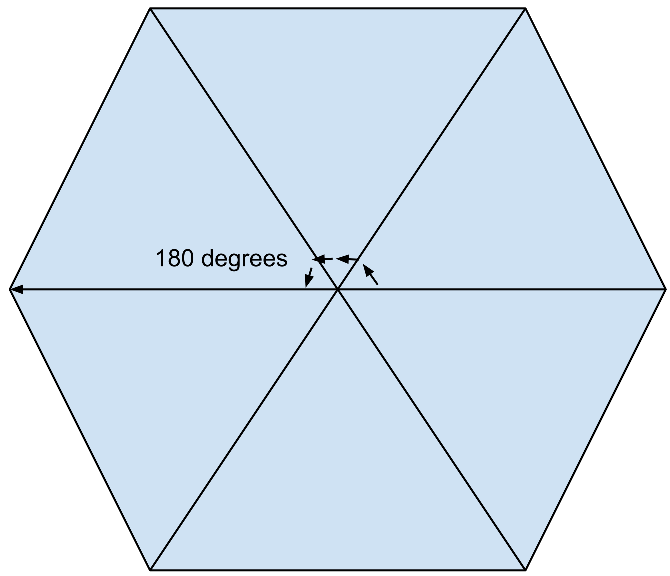
Let's start off with the "Motor Angle." The "Motor Angle" refers to the angle Motor 1 forms relative to the robot. We know that an equilateral hexagon can be divided into 6 equilateral triangles. Since the sum of all angles should add up to 360 degrees, dividing it by 6 will give us 60 degrees (the angle each triangle forms). This will take some time, but you will realize that the "Motor 1 Speed" vector lines up perfectly with the arrow below, giving us 60 degrees for Motor 1's motor angle.



This is Motor 2's motor angle:



This is Motor 3's motor angle:



Now that we know Motor 1's motor angle, we can now calculate the "Angle of Direction." Since the x and y coordinates returned by the joystick form a triangle, we can use inverse tangent to calculate the "Angle of Direction."

$$Angle\ of\ Direction\ =\ tan^{-1}(\frac{Y_{joystick}}{X_{joystick}})$$

Where $Y_{joystick}$ and $X_{joystick}$ are the values returned by the joystick.
Plugging the equations for $Angle\ of\ Direction$ and $Motor\ Angle$ into:

$$theta\ =\ Motor\ Angle\ -\ Angle\ of\ Direction$$

Gives the equation:

$$theta\ =\ 60^{\circ}\ -\ tan^{-1}(\frac{Y_{joystick}}{X_{joystick}})$$

And since we want to express $theta$ in radians, not degrees (we can only program in radians), we can multiply 60° by $\frac{\pi}{180}$ (degrees cancel out and we are left with radians). We do not need to convert $tan^{-1}(\frac{Y_{joystick}}{X_{joystick}})$ into radians as the $tan^{-1}()$ function provided by the Arduino library already returns the angle in terms of radians.

This thus gives us:

$$theta = (60^{\circ} * \frac{\pi}{180}) - tan^{-1}(\frac{Y_{joystick}}{X_{joystick}})$$

Finally, we put everything together. Here are the main equations we currently have:

$$Vspeed = cos(theta) * controller\ input$$

$$controller\ input = \sqrt{(Xval)^2 + (Yval)^2}$$

$$theta = (60^{\circ} * \frac{\pi}{180}) - tan^{-1}(\frac{Y_{joystick}}{X_{joystick}})$$

Putting them all together gives us:

Motor 1:

$$Vspeed = cos[(60^{\circ} * \frac{\pi}{180}) - tan^{-1}(\frac{Y_{joystick}}{X_{joystick}})] * [\sqrt{(Xval)^2 + (Yval)^2}]$$

Motor 2:

$$Vspeed = cos[(300^{\circ} * \frac{\pi}{180}) - tan^{-1}(\frac{Y_{joystick}}{X_{joystick}})] * [\sqrt{(Xval)^2 + (Yval)^2}]$$

Motor 3:

$$Vspeed = cos[(180^{\circ} * \frac{\pi}{180}) - tan^{-1}(\frac{Y_{joystick}}{X_{joystick}})] * [\sqrt{(Xval)^2 + (Yval)^2}]$$