# EQUATIONS FOR SWERVE DRIVE

The equation for the vertical and horizontal velocity of each swerve drive given the vertical, horizontal and angular velocity is given by this matrix:

$$\begin{bmatrix} v_{fr_x} \\ v_{fr_y} \\ v_{fl_x} \\ v_{fl_y} \\ v_{rl_x} \\ v_{rl_y} \\ v_{rr_x} \\ v_{rr_y} \end{bmatrix} = \begin{bmatrix} 1 & 0 & r_x \\ 0 & 1 & r_y \\ 1 & 0 & -r_x \\ 0 & 1 & r_y \\ 1 & 0 & -r_x \\ 0 & 1 & -r_y \\ 1 & 0 & r_x \\ 0 & 1 & -r_y \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix}$$

These are the parameters needed to calculate the angle and velocity of each swerve drive module:

- Vx is the linear component along the x-axis (horizontal component)
- Vy is the linear component along the y-axis (forward component)
- Wz is the angular velocity about the z-axis (yaw)
- Rx is the horizontal distance between the robot center and wheel pivot
- Ry is the vertical distance between the robot center and wheel pivot

Using this matrix to calculate the velocity of each wheel, you get:
Vfrx = Vx + Rx * Wz
Vfry = Vy + Ry * Wz

Vflx = Vx - Rx * Wz
Vfly = Vy + Ry * Wz

Vblx = Vx - Rx * Wz
Vbly = Vy - Ry * Wz

Vbrx = Vx + Rx * Wz
Vbry = Vy - Ry * Wz

The resultant linear velocity at the wheel is given by the vector sum of the robot's linear velocity ( $v_x$ & $v_y$ ) and tangential component of the angular velocity ( $\omega_z$ ).

$$\overrightarrow{v_w} = \overrightarrow{v} + \overrightarrow{r\omega_z}$$

Considering the front right wheel, breaking down the resultant linear velocity in x & y components:

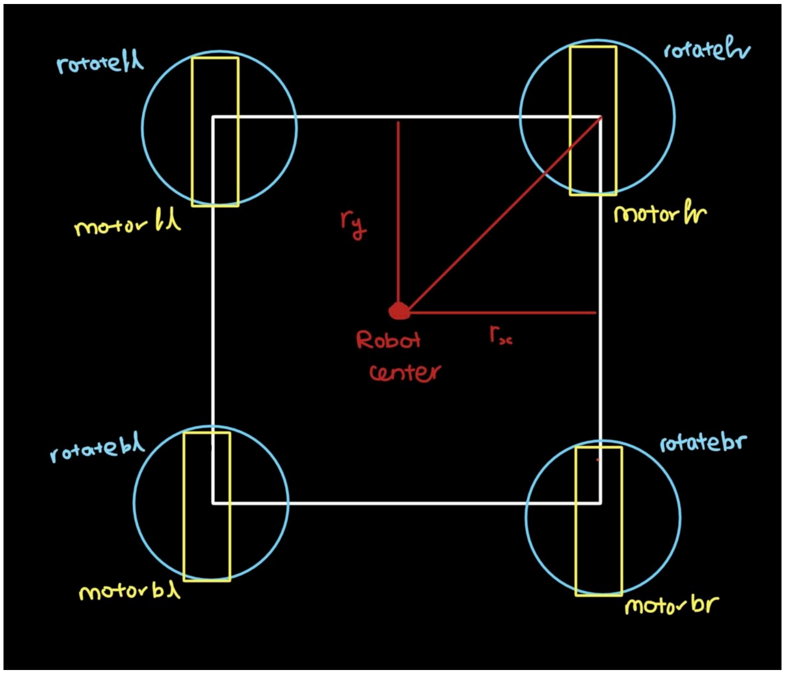
$$v_{fr_x} = v_x + r_x\omega_z$$

$$v_{fr_y} = v_y + r_y\omega_z$$

After obtaining the horizontal and vertical linear velocity of each individual swerve drive module, the angle and velocity of each swerve drive module can be calculated.

$$Steering\ Angle\ \Phi_{fr}\ (rad) = \tan^{-1}(\frac{v_{fr_y}}{v_{fr_x}})$$

$$Wheel\ Angular\ Velocity\ \omega_{fr}\ (rad/s) = \frac{\sqrt{v_{fr_x}^2 + v_{fr_y}^2}}{radius_{wheel}}$$




In the code, the steering wheels are labelled rotatefl, rotatefr, rotatebl and rotatebr and the movement wheels are labelled motorfl, motorfr, motorbl and motorbr.

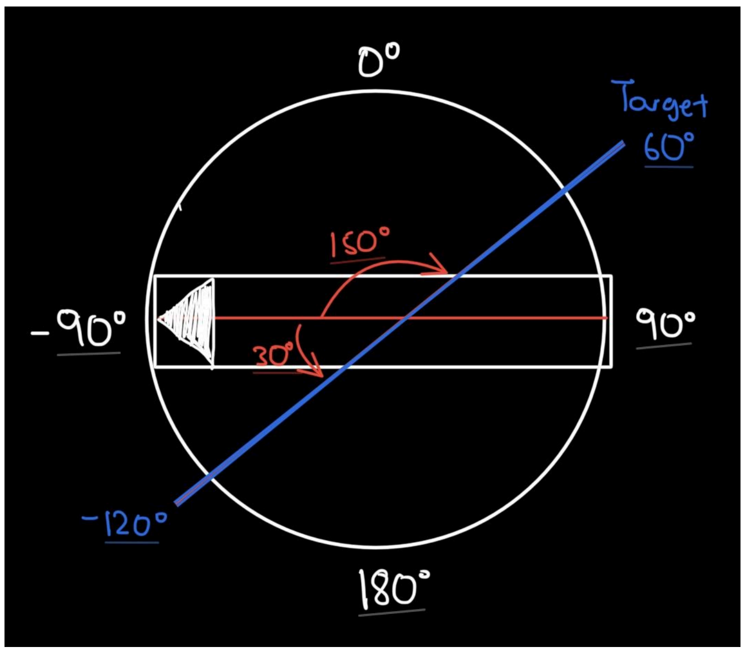
Rx is motor_x_dist and Ry is motor_y_dist, which will be normalized to be smaller than one in the code.

# ANGLE OPTIMISATION

Since the steering angle calculated will be between -90 degrees and 90 degrees (exclusive), and the wheel angular velocity is always positive, the robot will not be able to travel backwards with these calculations.

If the wheel is going backwards, its linear velocity in the forward direction would be negative, and we can flip the direction the wheel is travelling in by making the wheel angular velocity negative.

The steering angle can be optimised further taking into account that the wheel does not need to rotate more than 90 degrees in order to point in any direction.




For instance, assuming that the angle of the steering motor ranges from -180 degrees to 180 degrees and the steering motor is currently at the -90 degrees position, if the target angle is 60 degrees, it can either turn 150 degrees in the clockwise direction or 30 degrees in the anticlockwise direction. Turning 30 degrees anticlockwise would be preferred.

Hence, by adding 180 degrees to the motor's current angle, it will be normalized in the angle_pid() function to lie between -180 degrees and 180 degrees. The final target angle will then be (60 + 180 - 360) = -120 degrees, which causes the motor to turn 30 degrees.

The steering wheel will now be pointing in the opposite direction to the calculations, and the linear velocity of the wheel also has to be flipped by multiplying it by -1.