Simplistic Control of Mecanum Drive

by: Ian McInerney, FRC Team 2022

Overview:

Mecanum drive is a type of holonomic drive base; meaning that it applies the force of the wheel at a 45° angle to the robot instead of on one of its axes. By applying the force at an angle to the robot, you can vary the magnitude of the force vectors to gain translational control of the robot; aka, the robot can move in any direction while keeping the front of the robot in a constant compass direction. This differs from the basic robot drive systems like arcade drive, tank drive, or shopping cart drive require you to turn the front of the robot to travel in another direction.

Control Algorithm:

To get the most out of a mecanum drive system you will need to have the following information available to control it:

- Desired Angle – What angle the robot needs to translate at
- Desired Magnitude – What speed the robot must move at
- Desired Rotation – How quickly to change the direction the robot faces

Using that information the voltage multipliers for each wheel (numbers shown in Figure 1) can be calculated using the formulas found in Equation 1.

$$V_1 = V_d \sin(\theta_d + \frac{\pi}{4}) + V_\theta$$

$$V_2 = V_d \cos(\theta_d + \frac{\pi}{4}) - V_\theta$$

$$V_3 = V_d \cos(\theta_d + \frac{\pi}{4}) + V_\theta$$

$$V_4 = V_d \sin(\theta_d + \frac{\pi}{4}) - V_\theta$$

Equation 1: Formulas for calculating wheel speed

$V_x = \textit{The voltage multiplier for the } x^{th} \textit{ wheel}$

$V_d = \textit{Desired robot speed } [-1,1]$

$\theta_d = \textit{Desired robot angle } [0,2\pi]$

$V_\theta = \textit{Desired speed for changing direction} [-1,1]$

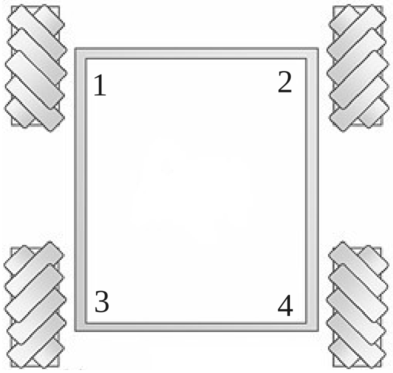


Figure 1: Wheel Numbering

These equations are composed of 2 main parts:

- Force vector computation

$$V_d \sin(\theta_d + \frac{\pi}{4})$$

- Modification for rotation

$$V_\theta$$

This value can be modified by a scaler quantity to make the rotation be more sensitive.

These equations will compute the voltage multiplier for the wheels, but they return a value which is between [-2,2]. This range is larger than the desired range of [-1,1]. The best way to fix that is to scale the outputs if any of them get above the desired range. This way the motors will travel at full speed in translation mode, but when rotation is commanded the outputs will scale to still be in the desired range of [-1,1] The process for scaling the motor outputs is shown in Chart 1.

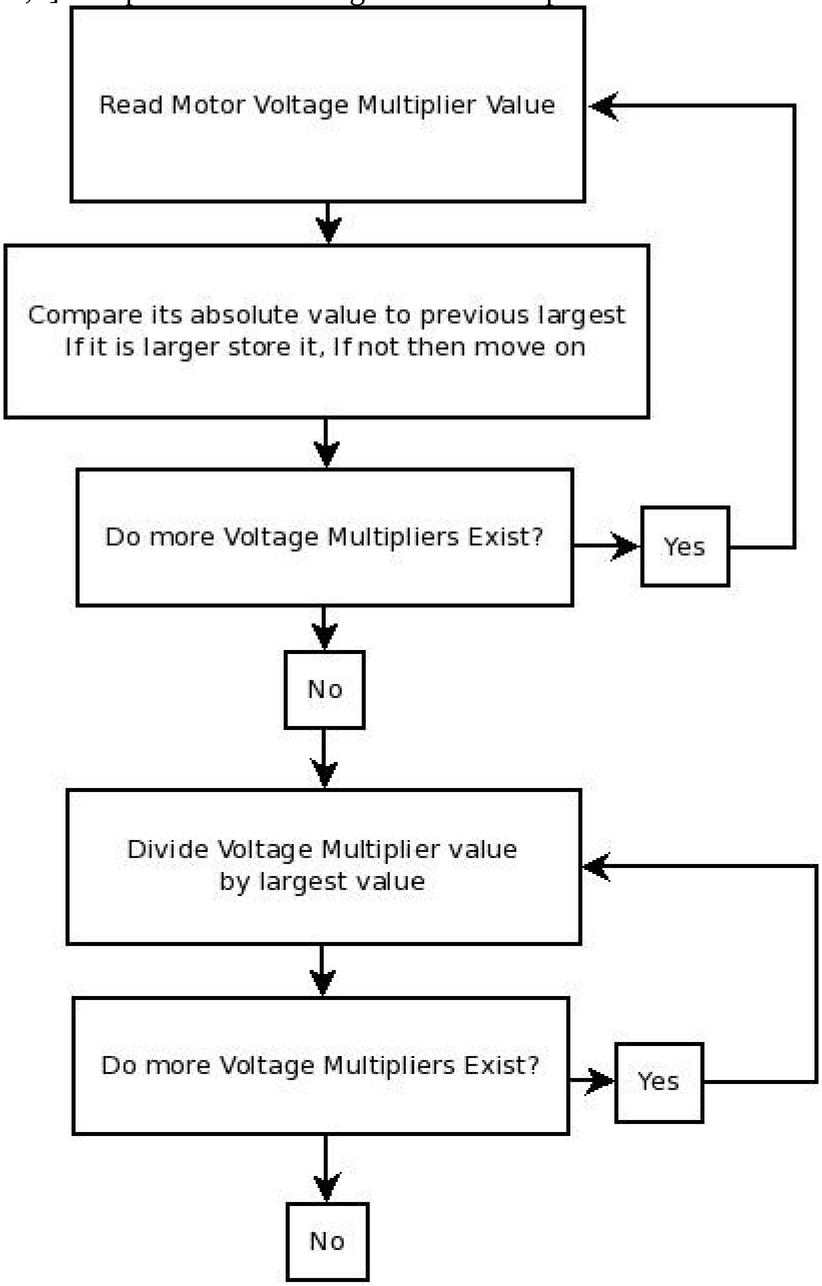

Read Motor Voltage Multiplier Value
Compare its absolute value to previous largest
If it is larger store it, If not then move on
Do more Voltage Multipliers Exist?
Yes
No
Divide Voltage Multiplier value
by largest value
Do more Voltage Multipliers Exist?
Yes
No

Chart 1: Normalization process for motor outputs

By using this algorithm to scale the motor outputs when they go outside the range [-1,1] the ratio between the motor speeds is preserved, maintaining the desired final force vector. This process also has the advantage that when full power is demanded in translation mode, the wheels will output full power, whereas if any of the values were to be modified using a scaler in the formulas in Equation 1 then full power would not be provided when commanded.

On the other hand, if a simple limiting function (See Equation 2 for an example of a piece-wise function) is used as the primary scaling method the final force vector will be different than the desired force vector since the ratio between the wheel speeds is not kept.

$$V(x)=\left\{\begin{matrix}-1 & \mathit{if}\ x\leqslant -1\\ x & \mathit{if}\ -1<x<1\\ 1 & \mathit{if}\ \leqslant 1\end{matrix}\right\}$$

Equation 2: Piece-wise defined function for limiting

Derivation of Algorithm:

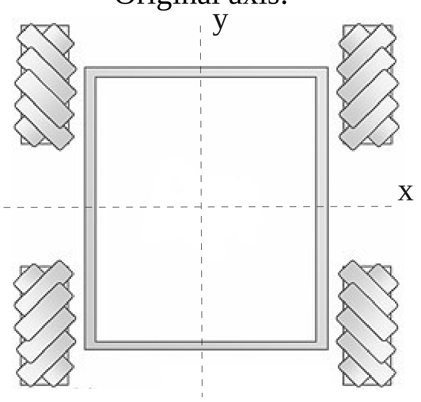
The above equations can be derived following the following steps:

1. Assume the following variables:

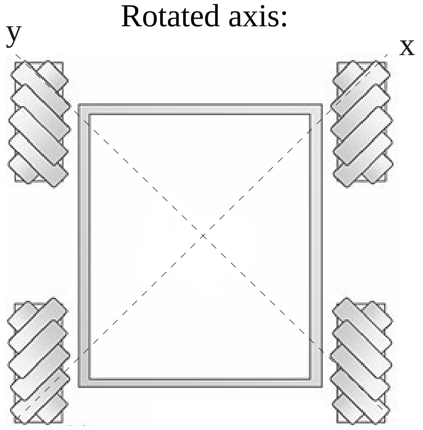
$$\begin{aligned}&V_x = \mathit{The\,voltage\,multiplier\,for\,the}\ x^{th}\ \mathit{wheel}\\ &V_d = \mathit{Desired\ robot\ speed}\ [-1,1]\\ &x_d = X\ \mathit{component\ of\ desired\ force\ vector}\\ &y_d = Y\ \mathit{component\ of\ desired\ force\ vector}\\ &\theta_d = \mathit{Desired\ robot\ angle}\ [0,2\pi]\\ &V_\theta = \mathit{Desired\ speed\ for\ changing\ direction}[-1,1]\end{aligned}$$

2. Rotate the axis of the robot 45° from normal.

Original axis:

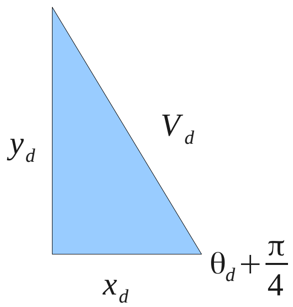


Rotated axis:



This makes the beginning of the equation: $\theta_d+\frac{\pi}{4}$

3. Determine the components for the desired force vector (on the new axis) using the desired force vector as the hypotenuse of a triangle



Then, using trigonometric identities you get the following components for the overall force vector

$$x_d=V_d\cos(\theta_d+\frac{\pi}{4})$$

$$y_d=V_d\sin(\theta_d+\frac{\pi}{4})$$

4. These components can then be assigned to specific wheels based upon the direction of the rollers on the wheels.
   Wheel 1 (Front Left) has rollers going in the y-direction, so it gets assigned the y-component
   Wheel 2 (Front Right) has rollers going in the x-direction, so it gets assigned the x-component
   Wheel 3 (Rear Left) has rollers going in the x-direction, so it gets assigned the x-component
   Wheel 4 (Rear Right) has rollers going in the y-direction, so it gets assigned the y-component

This creates the following 4 equations

$$V_1=V_d\sin(\theta_d+\frac{\pi}{4})$$

$$V_2=V_d\cos(\theta_d+\frac{\pi}{4})$$

$$V_3=V_d\cos(\theta_d+\frac{\pi}{4})$$

$$V_4=V_d\sin(\theta_d+\frac{\pi}{4})$$

5. These equations will work for just translating the robot, but to be able to change the direction faced the rotation value needs to be added to the right wheel values and subtracted from the left wheel values. You can then adjust the sensitivity of the rotation by scaling V_θ appropriately.

This gives the following equations:

$$V_1=V_d\sin(\theta_d+\frac{\pi}{4})+V_\theta$$

$$V_2=V_d\cos(\theta_d+\frac{\pi}{4})-V_\theta$$

$$V_3=V_d\cos(\theta_d+\frac{\pi}{4})+V_\theta$$

$$V_4=V_d\sin(\theta_d+\frac{\pi}{4})-V_\theta$$

Implementation:

This paper will not focus on individual implementation of the algorithm, it will just provide a high-level overview. The high-level block diagram can be found in Chart 2.

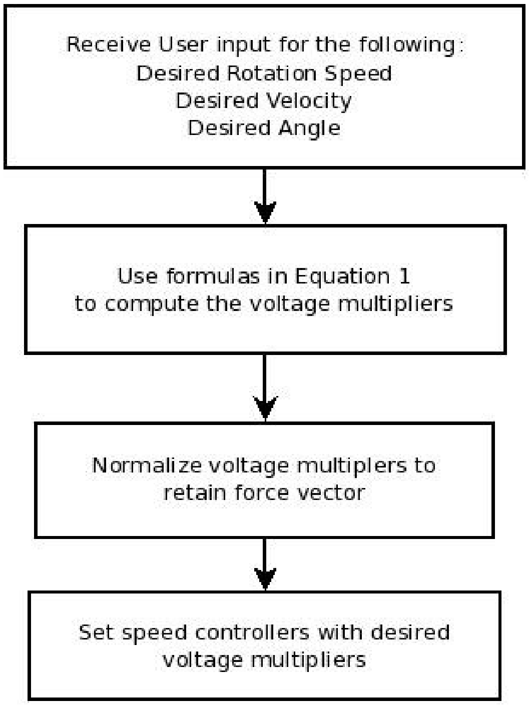

Receive User input for the following:
Desired Rotation Speed
Desired Velocity
Desired Angle
Use formulas in Equation 1
to compute the voltage multipliers
Normalize voltage multiplers to
retain force vector
Set speed controllers with desired
voltage multipliers

Chart 2: High-level Block Diagram for control