Mecanum Drive Code

**Author**: Jason Bell

**When**: November 2, 2014

# About the Code

This code was written to abstract away the details in controlling a mecanum drive system. A MecanumDrive class is instantiated with a motor controller object and a simple call to the available functions let the user drive the vehicle without the need to interface directly with the motor controller.

**Drive System**

The system is simplified as follows. The user needs to only specify three things to drive the system with the MecanumDrive class.

1. The speed of the robot
2. The desired angle of translation
3. The desired speed of rotation

The speed of the robot is specified on the domain of [-1, 1]. The desired angle of translation is specified on the domain of [0, 2π]. The desired speed of rotation is specified on the domain of [-1, 1]. A diagram of the robot is shown in Figure 1 below. The diagram indicates the robots orientation in terms of angle of translation.

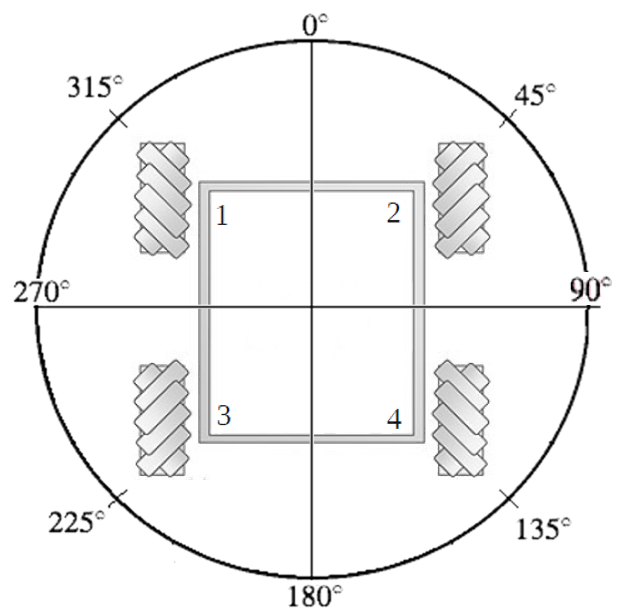
****

Figure 1. A diagram of the drive system.

**Equations**

The following equations are used in the MecanumDrive class to control the drive system. These equations are computed behind the scenes and are hidden from the user. The equations are taken from [1].

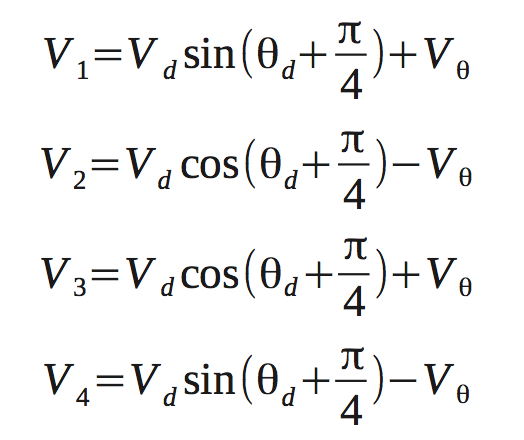
****

Figure 2. Equations used to drive the mecanum wheels.

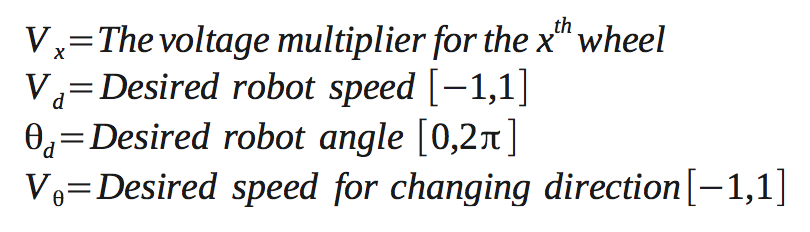
****

Figure 3. Parameter explanation.

**High Level Block Diagram**

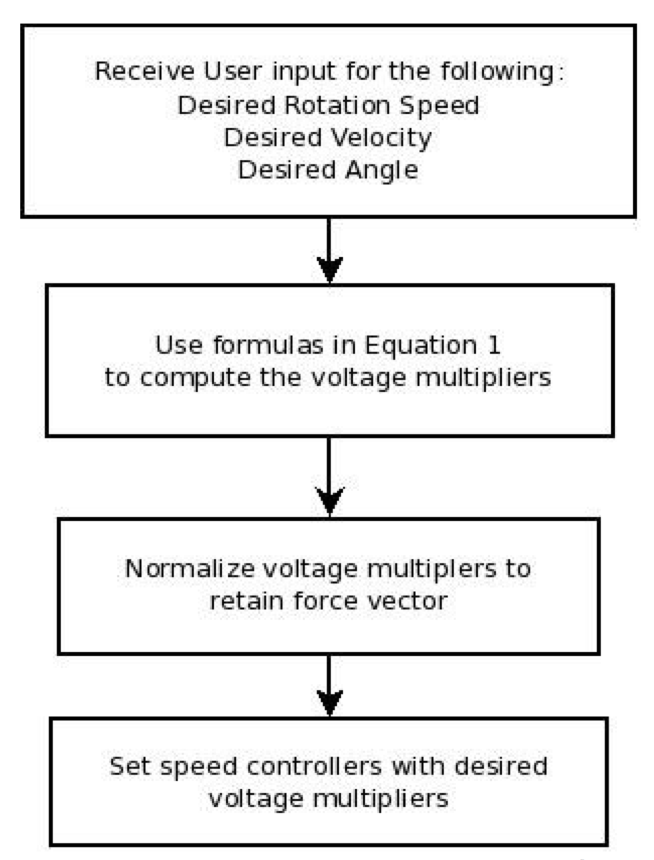
****

Figure 4. High level block diagram for the drive algorithm.

**Class Methods**

**parameters:**

velocity : the desired speed of translation [-1, 1]

degrees : the desired angle of translation in degrees [0, 360]

rotation : the desired speed of rotation [-1, 1]

voidinitialize ( )

* this function is used to initialize all components needed by the drive system including the motor controller and motor objects.

void setMaxSpeed( double speed)

* sets the maximum speed of the motors with percentage from 0 to 1

void drive( double velocity, int degrees, double rotation)

* drive with the given parameters

void stop()

* stop all movement

void forward ( )

* drive foward

void reverse ( )

* drive backward

void strafeLeft ( )

* move the robot leftward without rotating

void strafeRight ( )

* move the robot rightward without rotating

void rotateLeft ( )

* rotate the robot counter clockwise

void rotateRight ( )

* rotate the robot clockwise

**Usage**

To use the MecanumDrive code the user needs to first include the following header files: Wire.h, and MecanumDrive.h. After this is done the user can use the MecanumDrive namespace and the functions above to control the drive system. The driving functions will drive the motors at full speed by default. This behavior can be changed via the setMaxSpeed function. This function allows the user to give a percentage between 0 and 1 that will be used to control the speed of the motors. The only thing left to do is call one of the provided functions to engage the drive system. The available methods are described above. Several functions are provided for simple movement: forward, reverse, strafe, and rotate. If finer control is needed use the drive function. The drive method allows the user to specify proportional translational and rotational speeds along with translational angle. A simple example sketch showing use of the code is given below.

☞ The velocity and rotational speeds specified in the call to the drive method **do not** define the speed of the motors. These parameters are used in the equations are used to calculate raw values between -2 and 2. These values are then normalized to a -1 to 1 scale. When these new values are converted to a format used by the motor controller the values are scaled by the maxSpeed value.

**Example Sketch**

#include “MecanumDrive.h”

void setup()

{

MecanumDrive::initialize();

MecanumDrive::setMaxSpeed(.5) // sets the motors to operate at 50% speed

}

void loop()

{

// call the drive method

MecanumDrive::drive(1,90,0) // strafe right with at maxSpeed and no rotation

delay(10);

}

**References**

[1] Ian McInerney. “Simplistic Control of Mecanum Drive.” FRC Team 2022.