Drivetrain basics for programmers V1

\_\_\_\_\_\_ \_\_\_\_\_\_

| | \_\_\_\_\_ \_\_\_\_\_ | |

| |-|M(L1)| |M(R1)|-| |

|\_\_\_\_\_\_| |\_\_\_\_\_| |\_\_\_\_\_| |\_\_\_\_\_\_|

\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_

| | |M(L2)| |M(R3)| | |

| |-|\_\_\_\_\_| |\_\_\_\_\_|-| |

|\_\_\_\_\_\_| |\_\_\_\_\_\_|

This is a simple representation of the mecanum drivetrain in robotics.

A mecanum drivetrain uses 4 motors 2 on each side to move forward and backwards, turn

and unique to this drivetrain, strafing or in other words going from left to right without turning (as expressed below).

\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_

| | | | | |

| | <-- | | --> | |

|\_\_\_\_\_\_| |\_\_\_\_\_\_| |\_\_\_\_\_\_|

Strafing is sometimes difficult to code and I hope that I can work on explaining the basics in this guide.

to start however, we need to go over the basics of controlling the chassis that works for both mecanum and six wheel.

the first thing that you want to do is make the chassis and when your done, make a mapping on the rev driver hub.

once that is done, you need to take the control hub and connect it properly to your computer.

Now that the initial setup is done we will talk about how we can go about controlling our newly setup drivetrain.

1. tank drive. tank drive uses both joysticks to control the power of each side of the drivetrain i.e.: the left motors are controlled

by the input of the left joystick and the right side is controlled by the right joystick this is pretty straight forward so I will not

go into much detain about it

2. regular drive. the drivetrain turns based on how much you move the x axis on the left joystick and the robot goes forward and back

based on the y axis input of the left joystick. I will be going over this type exclusively in this guide.

the basic drivetrain controls for both mecanums and six wheel is:

1. Forward and back

2. turning left and right

1. Forward and back

this is very simple to do and should be the first thing that you do when you start to code.

to start using the rev library for ftc: Note: if you are using android studio the following imports will be imported for you (I recommend it):

package org.firstinspires.ftc.teamcode;

import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;

import com.qualcomm.robotcore.eventloop.opmode.TeleOp;

import com.qualcomm.robotcore.hardware.DcMotor;

once though are imported then you have to define the motors: Note: if you are planning to use your motors in more functions than they are defined in use this:

DcMotor left1;

DcMotor left2;

DcMotor right1;

DcMotor right2;

public void runOpMode() throws InterruptedException {

// sets the different motors

left1 = hardwareMap.get(DcMotor.class, "left1");

left2 = hardwareMap.get(DcMotor.class, "left2");

right1 = hardwareMap.get(DcMotor.class, "right1");

right2 = hardwareMap.get(DcMotor.class, "right2");

}

this code defines the motors but if you only want to use the motors in one function then just take out the definitions of the DcMotor variables and put them straight into the teleop function.

next, to make these motors spin forward we can execute the following command in the previous function.

spinleft((-1 \* gamepad1.left\_stick\_x));

//right

spinright(gamepad1.left\_stick\_x);

//left

and add the following functions:

public void spinleft ( double leftpower) {

// turns the 2 individual motors into a motor group

left1.setPower(leftpower);

left2.setPower(leftpower);

}

public void spinright ( double rightpower) {

// turns the 2 individual motors into a motor group ( right side )

right1.setPower(rightpower);

right2.setPower(rightpower);

}

note that tilting the joystick forward gives a negative value so we have to multiply it by another negative value to get the motors to go straight

I would recommend taking a look at the main. Java file I have on my GitHub for an more clean sample.

2. Turning

turning is more complicated because we have to tell the left and right sides to preform different actions based on the position of the joystick.

the different positions are as follows:

when we want to turn left, the left side has to be going backwards and the right side has to go forward

when we want to turn right, the left side has to go forwards while the right side goes backwards

from these statistics we can conclude that the side that we want to turn to has to be negative an the side we are turning away from has to be positive

and from that we can draw the following code:

left1.setPower(-1 \* gamepad1.left\_stick\_x);

left2.setPower(gamepad1.left\_stick\_x);

right1.setPower(gamepad1.left\_stick\_x);

right2.setPower(-1 \* gamepad1.left\_stick\_x);

!behold how the robot has turned!

I mentioned earlier a term called straifing. Strafing allows us to go left and right without turning. strafing allows for extreme maneuverability and was extremely helpful in last year’s game "CENTERSTAGE" and I hope that it might be helpful to your team this year.

if this is your mecanum drivetrain

\_\_\_\_\_\_ \_\_\_\_\_\_

| | \_\_\_\_\_ \_\_\_\_\_ | |

| |-|M(L1)| |M(R1)|-| |

|\_\_\_\_\_\_| |\_\_\_\_\_| |\_\_\_\_\_| |\_\_\_\_\_\_|

\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_

| | |M(L2)| |M(R3)| | |

| |-|\_\_\_\_\_| |\_\_\_\_\_|-| |

|\_\_\_\_\_\_| |\_\_\_\_\_\_|

then this is the motions required to straif left

\_\_\_\_\_\_ \_\_\_\_\_\_

|| | | \_\_\_\_\_ \_\_\_\_\_ | | /\

|| | |-|M(L1)| |M(R1)|-| | ||

\/ |\_\_\_\_\_\_| |\_\_\_\_\_| |\_\_\_\_\_| |\_\_\_\_\_\_| ||

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/ The drivetrain moves left |

\\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_

/\ | | |M(L2)| |M(R3)| | | ||

|| | |-|\_\_\_\_\_| |\_\_\_\_\_|-| | ||

|| |\_\_\_\_\_\_| |\_\_\_\_\_\_| \/

and then finally this is the motor config to straif right

\_\_\_\_\_\_ \_\_\_\_\_\_

/\ | | \_\_\_\_\_ \_\_\_\_\_ | | ||

|| | |-|M(L1)| |M(R1)|-| | ||

|| |\_\_\_\_\_\_| |\_\_\_\_\_| |\_\_\_\_\_| |\_\_\_\_\_\_| \/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| The drivetrain moves right \

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_

|| | | |M(L2)| |M(R3)| | | /\

|| | |-|\_\_\_\_\_| |\_\_\_\_\_|-| | ||

\/ |\_\_\_\_\_\_| |\_\_\_\_\_\_| ||

in straifing, both sides send apposing forces which cancel out but because the wheels

are made of wheels when this happens the side that is moving inward, pulls the robot towards

it's side which is how straifing works!

now, to code this all we have to do is make the 2 bumpers the activation so if the left bumper is

pressed it straifs to the right but if the right bumper is pressed it straifs to the right.

therefore we can insert the following code:

strafeleft(gamepad1.left\_bumper);

straferight(gamepad1.right\_bumper);

and insert the following functions:

public void strafeleft (boolean straifleft) {

boolean True = true;

if (straifleft == True); {

straifmotion(1);

}

}

public void straferight (boolean straifright) {

boolean True = true;

if (straifright == True); {

straifmotion(1);

}

}

// note this function should only be used with -1, or 1

// the defalt way is also left

public void straifmotion (int posorneg ) {

left1.setPower(-1 \* posorneg);

left2.setPower(posorneg);

right1.setPower(posorneg);

right2.setPower(-1 \* posorneg);

}

and that is it you have learned the basics of operating drivetrains (mecanum and six wheel).