Finer Velocity Control

Detecting a Large Change in Velocity

We need to put a limit on how much our robot's velocity can change at any given time. We will do this using a proportional controller.

Our change in pitch motor velocity (ΔY) is given by the current velocity (YM) minus the new target velocity (YN).

$$\Delta Y = YM - YN$$

From the previous document, if this difference is greater than positive or negative 180, our arm will break. This gives us the following two conditions:

$$YM - YN \leq 180$$

$$YM - YN \geq -180$$

If both of these conditions are true, then our robot can safely change the velocity of its yaw motor to YN. If either of these equations is false, then changing the velocity of the yaw motor to YN will result in a change of velocity of more than ± 180 and will break the arm. We will therefore need to calculate a different velocity to change to.

How do we know what the current yaw velocity (YM) is? Unfortunately, we don't have a sensor that tells us what our current yaw or pitch velocities are. We will need to keep track of these values ourselves. We can keep track of the current velocities by remembering the last values that we assigned to our motors. We store these values in variables.

Variables are containers for storing data. There are two main ways to interact with variables: assigning data to the variable and retrieving the data stored in the variable. In this Project, we have access to two variables: Current Pitch Speed and Current Yaw Speed. Here is how to use these variables in JavaScript:

Unfortunately, we can't declare and update our own custom variables in the text editor. This is because your code is being run inside an invisible outer while loop. The code that you write in the editor is being constantly re-run multiple times a second. So, if you declare a variable in your code, the value of that variable will constantly be reset each time your code is re-run.

Confusing? Don't worry about it. The only thing that you need to know is that to keep track of a variable between iterations of your code, you will need to use one of our predefined functions.

We have already declared a variable called CurrentYawSpeed outside of this invisible while loop. You can assign values to this variable using a function called setCurrentYawSpeed(number) and you can retrieve the value of this variable using a function called getCurrentYawSpeed().

For example, if our robot's current yaw speed is 5 and we want to remember this for our next iteration, we can store it using a function call like so:

```
setCurrentYawSpeed(5);
```

We can then retrieve this value in our next iteration with the following function call:

```
getCurrentYawSpeed();
```

In the example below, we use the getCurrentYawSpeed() function to retrieve the value of the current yaw speed variable. We then add one to this value and assign it back to the variable using the setCurrentYawSpeed() function. We then print the value of the variable to the console before iterating again after a 1-second delay.

```
let out_0;

out_0 = getCurrentYawSpeed();
setCurrentYawSpeed(out_0 + 1);
console.log(getCurrentYawSpeed())

Delay(1)
```

If we run the code above, we will see the value of the current yaw speed variable increase by one during each iteration. This corresponds to the following logic:

```
Let Current_Yaw_Speed = 0
While True
    Current_Yaw_Speed = Current_Yaw_Speed + 1
    Print Current_Yaw_Speed
End
```

How can we use these variables to build our solution? If we want to keep track of the yaw velocity, we can assign the same value that we assign to the yawSpeed() function to the setCurrentYawSpeed() function. Then, whenever we need to use this value elsewhere in our code, we can add the getCurrentYawSpeed() function to our code to access the latest value assigned to this variable.

Velocity Limits

If we detect that our change in velocity is going to be more than ± 180, we will need to reduce our target speed to be closer to our current speed. At the same time, however, we still want to be moving as fast as possible. If our maximum change in velocity is ± 180, then if we exceed that number we could just reduce it down to be exactly ± 180. We could do this using for loops, however, our code might get quite complicated. Another solution would be to use the JavaScript equivalent of a clamp block.

A clamp block takes any numbers as its three inputs, performs a calculation, then outputs a new number. We will refer to these inputs as input a, input b and input c. The calculations are as follows:

1. If a is less than b, then set a to be equal to b
2. Else if a is greater than c, then set a to be equal to c
3. If a is between b and c (inclusive), then leave it alone
4. Then return the current value of a

Or in other words:

```
If a < b
        a = b
Else if a > c
        a = c
Return a
```

We can replicate this functionality in JavaScript using a combination of a Math.min() and a Math.max() function.

The function Math.min() returns the number passed into it with the lowest value. Conversely, Math.max() returns the number passed into it with the highest value.

In the following code, we get the current yaw angle, add one to it, get the maximum between 0 and that number, get the minimum between 10 and that number, then set the current yaw speed to this new value. This piece of code effectively sets the yaw speed to be equal to the current yaw angle plus one, down to a minimum of 0 and up to a maximum of 10. Check if this matches our definition of a clamp function above.

```
let out;

out = Math.min(Math.max(yaw + 1, 0), 10)
console.log(out)
yawSpeed(out)

Delay(1)
```

Think about how the clamp functionality could be used to make sure that our change in velocity remains greater than or equal to -180 but less than or equal to 180.

Putting It All Together

1. How do we detect if the change of velocity is going to be too large?
    a. If the difference between the current velocity and the target velocity is less than -180 or greater than 180.
2. If the change in velocity is going to be fine, how do we change the velocity?
    a. The same as the previous subsystems, likely the difference in the target angle and the current angle multiplied by some constant found through experimentation.
3. If the change in velocity is going to be too large, what do we change the velocity to instead?
    a. -180 or 180, whatever is closer.