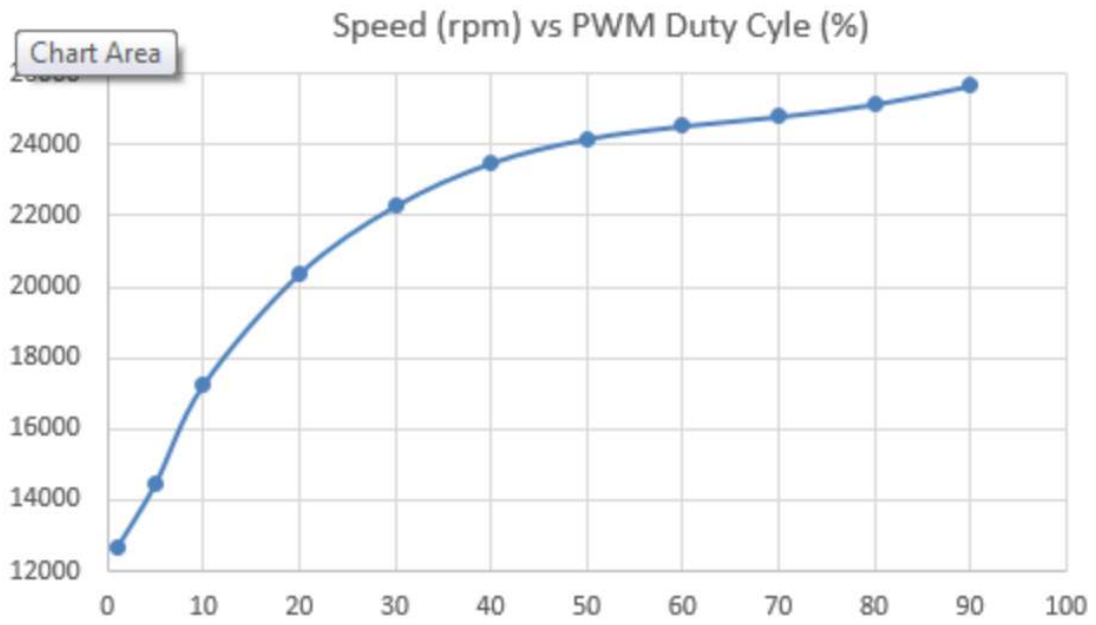
Tutorial 6 Actuators Solutions

Wednesday, 27 August 2025 11:43 AM

Question 3

Provide a software solution. By idle speed, he means no load speed

---



So we tabulate these values.

To make life easier, we only tabulate every 1000 velocity increases.

So we have

Array Index	Velocity (RPM)	PWM Duty Cycle (%)
0	0	0
1	13000	1
2	14000	4
3	15000	7
4	16000	8
5	17000	10
6	18000	12
7	19000	15
8	20000	19
9	21000	24
10	22000	28
11	23000	35
12	24000	46
13	25000	75

Pseudo code

```
int desiredSpeed = (insert desired speed)

lookupArray = (insert table from before as an array, where the value is the PWM duty cycle %)

if (desiredSpeed > 25000 || desiredSpeed < 13000) -> invalid argument, try again

int leftDigits = desiredSpeed / 1000

int rightDigits = desiredSpeed % 1000

arrayIndex = leftDigits - 12

interpolationPercentage = rightDigits / 1000
interpolationSlope = lookupArray[arrayIndex + 1] - lookupArray[arrayIndex]
interpolationValue = interpolationSlope * interpolationPercentage

finalPwmValue = lookupArray[arrayIndex] + interpolationValue
```

Take 21435 as an example value

Left digit is 21
Right digit is 435
Array index is 9
Interpolation percentage is 0.435
Interpolation slope is 28 - 24 = 4
Interpolation value is 4 * 0.435 = 1.74
Final PWM Value is 24 + 1.74 = 25.74

So this is better than just rounding and just using the array value
However, we are interpolating which assumes a linear relationship between the values. We can see on the graph that this is incorrect. However, it's still better than nothing.

The more we sample, the more array values we have, the better our interpolations or guesses become. Of course, the more we sample the more you could argue that we just obtain the formula of the curve. This is the optimal solution, but can be hard, so interpolation provides an "easy way out" while still having decent performance/accuracy.