RBE2002 Review Session Answers

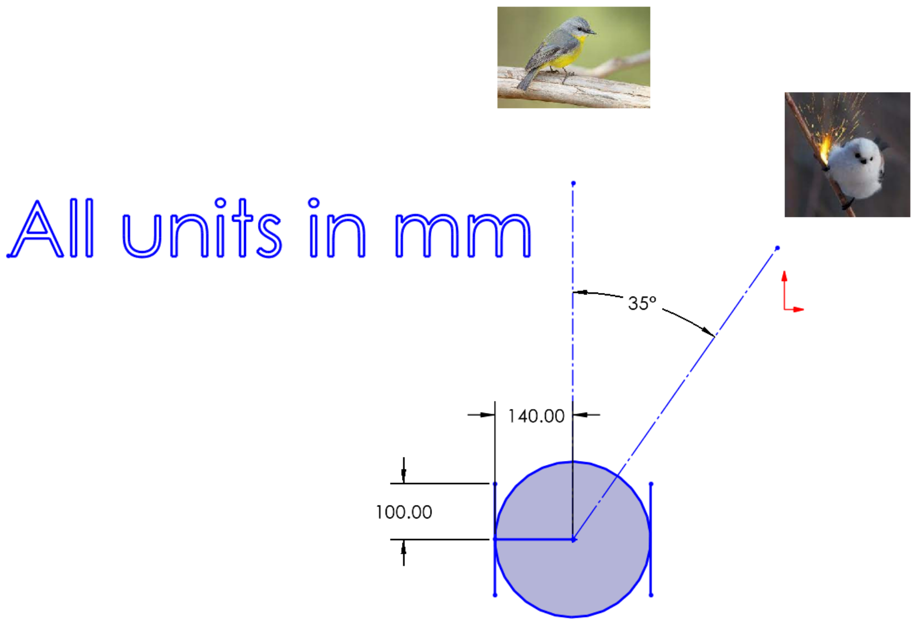
November 20th 2023

1. Using C++, calculate the ICC position given the romi wheel speeds, wheel track, and pose. Then print it out.

```cpp
// Wheel Track in mm
const float w_track = 200;

// These variables are being updated elsewhere, you can use them as
    is
// Current x (in m), current y (in m), current theta (in deg)
float curr_x, curr_y, curr_th;

// Calculate ICC position from Romi pose and wheel speeds
void print_ICC(float speed_l, float speed_r)
{
    if (speed_l == speed_r)
    {
        Serial.println("speeds too close");
    }
    else
    {
        float diff_speeds = speed_l - speed_r;
        float w_track_new = w_track * 1 / 1000; // go from mm to m
        float R = w_track_new / 2 * (speed_r + speed_l) / diff_speeds;
        float th_rad = curr_th * 2 * pi / 360; // go from deg to rad
        float ICCx = curr_x - R * sin(th_rad);
        float ICCy = curr_y + R * cos(th_rad);

        Serial.print(ICCx);
        Serial.print('\t');
        Serial.println(ICCy);
    }
}
```

---

2. You have your romi set up to look at cool birds. You know there's a cooler bird 35 degrees to the right of where the romi is currently looking. The romi has a 35:1 gearbox. One revolution of the encoders magnetic disk is 700 counts. How many encoder counts does each wheel have to spin to do a point turn to look at the cooler bird?



Answer:

Lets first calculate how much the wheel goes:

$$romiRadius = 140mm$$

So, our wheels will go forward:

$$wheelDist = 35\deg/360\deg * 2\pi * romiRadius = 85.5mm$$

So, our wheel has to turn 85.5mm, then we can find how many degrees does the wheel need to spin:

$$wheelRadius = 100mm$$

So, our wheels will go forward:

$$\theta/360deg * 2\pi * wheelRadius = wheelDist$$

$$\theta = wheelDist * 360deg/(2\pi * wheelRadius)$$

$$\theta = 49deg$$

Now we can find the amount of revolutions our wheels need to do:

$$revolutions = \theta/360deg = 0.136rev$$

With the gearbox we get the revolutions of the motor:

$$motorRev = revolutions * 35 = 4.76revsmotor$$

And finally we get our encoder counts:

$$motorRev * 700counts/1rev = 3334counts$$

3. When powered by 5v, the photosensor has a transfer function of

$$Vout = 1.5V + 15mV * L$$

Where Vout is in volts and L is light in lux. You intend to sample the sensor with an ADC that will have a reference voltage of 3.3V. What is the max lux that can be registered?

Answer:

So, we have a reference voltage:

$$Vref = 3.3V$$

At maximum lux, our Vout will be the same as Vref, so:

$$Vref = 1.5V + 0.015V * L$$

$$L = 1.8V/0.015V = 120lx$$

4. Consider another photosensor that has been placed in series with a 20k resistor. The system uses a reference voltage of 3.3v and the resistance of the photosensor can be described with the transfer function below.

$$Rf = 10k\Omega + 10\Omega * L$$

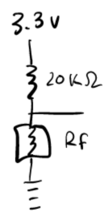
Where Rf is the resistance in ohms of the photosensor and L is the light sensed in lux. What is the voltage measured by the photosensor if it is given 200 lux?

Answer:

So, we first calculate Rf:

$$Rf = 10k\Omega + 10\Omega * 200 = 12k\Omega$$

So, then we have a simple voltage divider:



From which we can use our known equations to get Vout:

$$Vout = Vref * (Rf/(Rf + 20k\Omega))$$

$$Vout = 3.3V * (12/32)$$

$$Vout = 1.2375V$$

5. Using your answer from the last slide, what is the ADC value registered if a 10 bit ADC is attached to the photosensor?

Answer:

$$Vout = ADC * Vref/2^n$$

where

$$n = 10bits$$

So:

$$ADC = Vout/Vref * 2^n = 1.2375V/3.3V * 1024 = 384$$

Remember, here 1024 represent the amount of values we are able to represent.