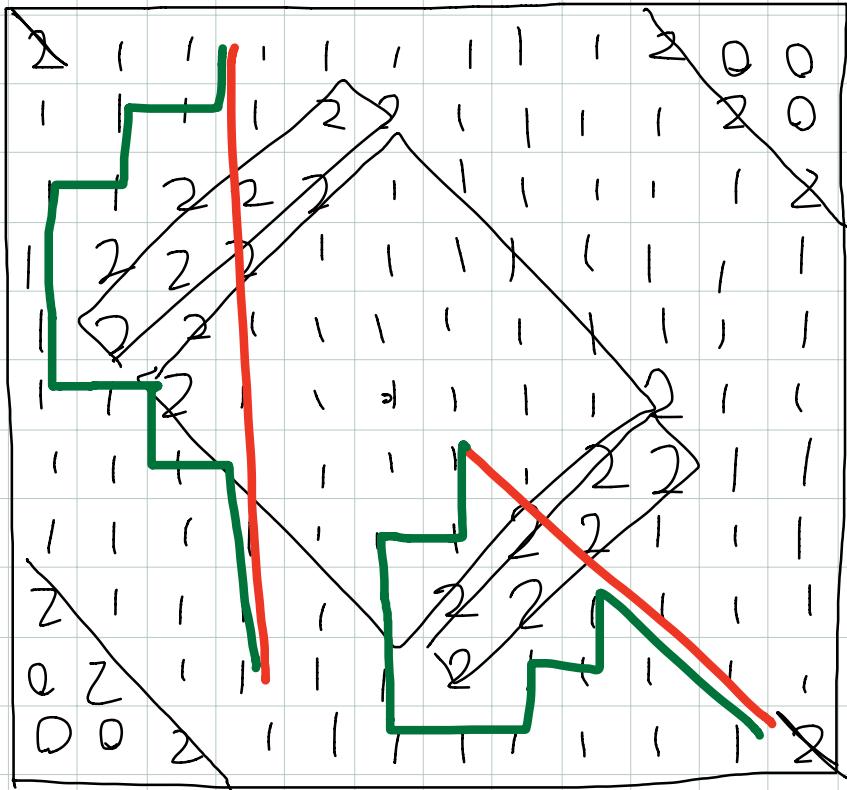


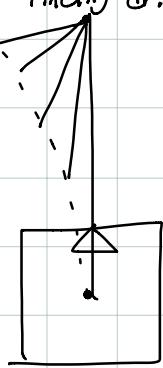
Example of field array



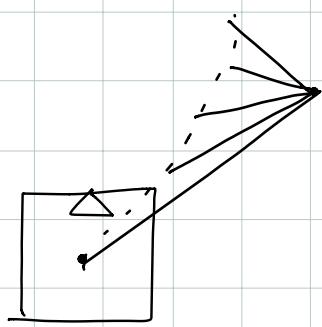
Things we must know how to do:

1. How to identify an index as 0, 1, or 2 (obstacle detection)
2. Differentiate between an obstacle and drift
3. What to do when an obstacle is found before and after field array is constructed
4. Maintain accurate location tracking on ramps - suspension + gyro
5. Pathing around known obstacles

②. In ← finding drift



current θ should = goal θ



Expected vs. Actual
expected Position vs. Current Position

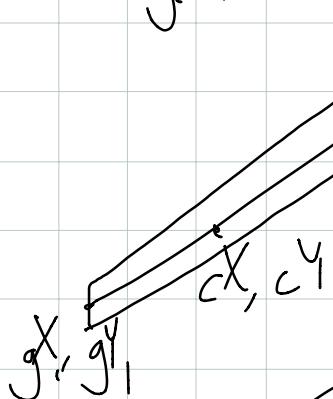
$$\theta \quad |goal\theta - current\theta| < 0.005$$

x
y



goal X
goal Y
goal θ

gX_2, gY_2



$$\frac{gY_2 - gY_1}{gX_2 - gX_1} (cX - gX_1) = (cY - gY_1)$$

Drift constant

$$\left| \frac{gY_2 - gY_1}{gX_2 - gX_1} (cX - gX_1) - (cY - gY_1) \right| < 0.05cm$$

① Sudden speed drop = obstacle hit

③ ~~Before array construction:~~

~~Back up, turn right, keep going~~

After array construction (most likely a robot has been encountered)

Back up,

use EOPD to find most open space,
drive right/left a little

drive in direction of EOPD for 18 in (unless obstacle encountered)

resume goToPosition commands

⑤ ~~if point along found to be a 2~~

Maybe the field array complicates things too much

Cons:

Need to create field array

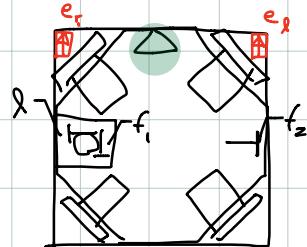
Doesn't help with robot detection

Pros:

It's cool

Is the building block to autonomous in teleOp

The new Entach robot:



- Tracks x , y , & θ using trackball and encoder
- Drift correction with expected vs. actual ($m(x - x_e) = y - y_e$)
- Speed trackers for all three encoders
obstacle detected when speed drops
- Two front EOPDs for obstacle detection and alternative path finding

Globals (or maybe statics)

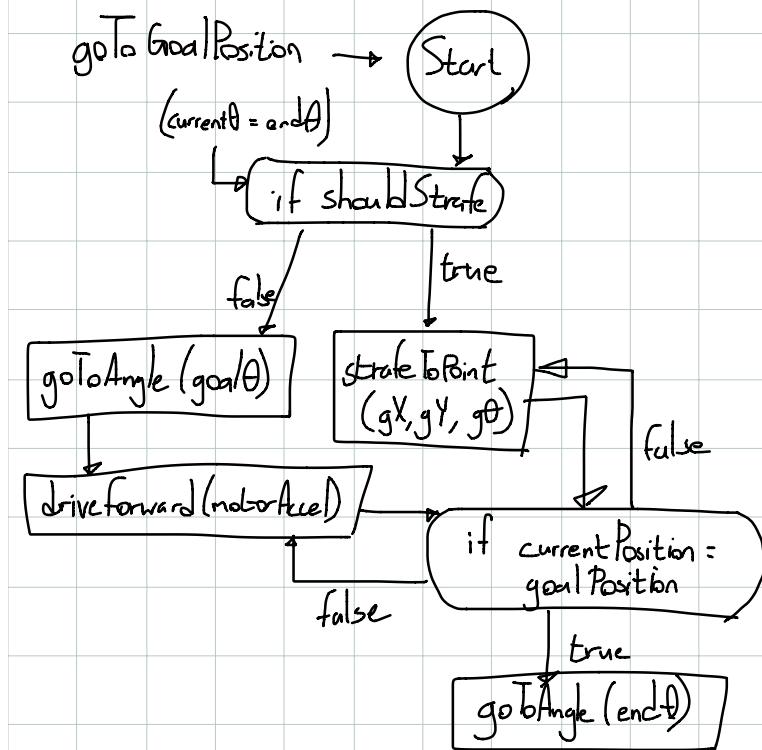
float goal X, Y, θ , end θ

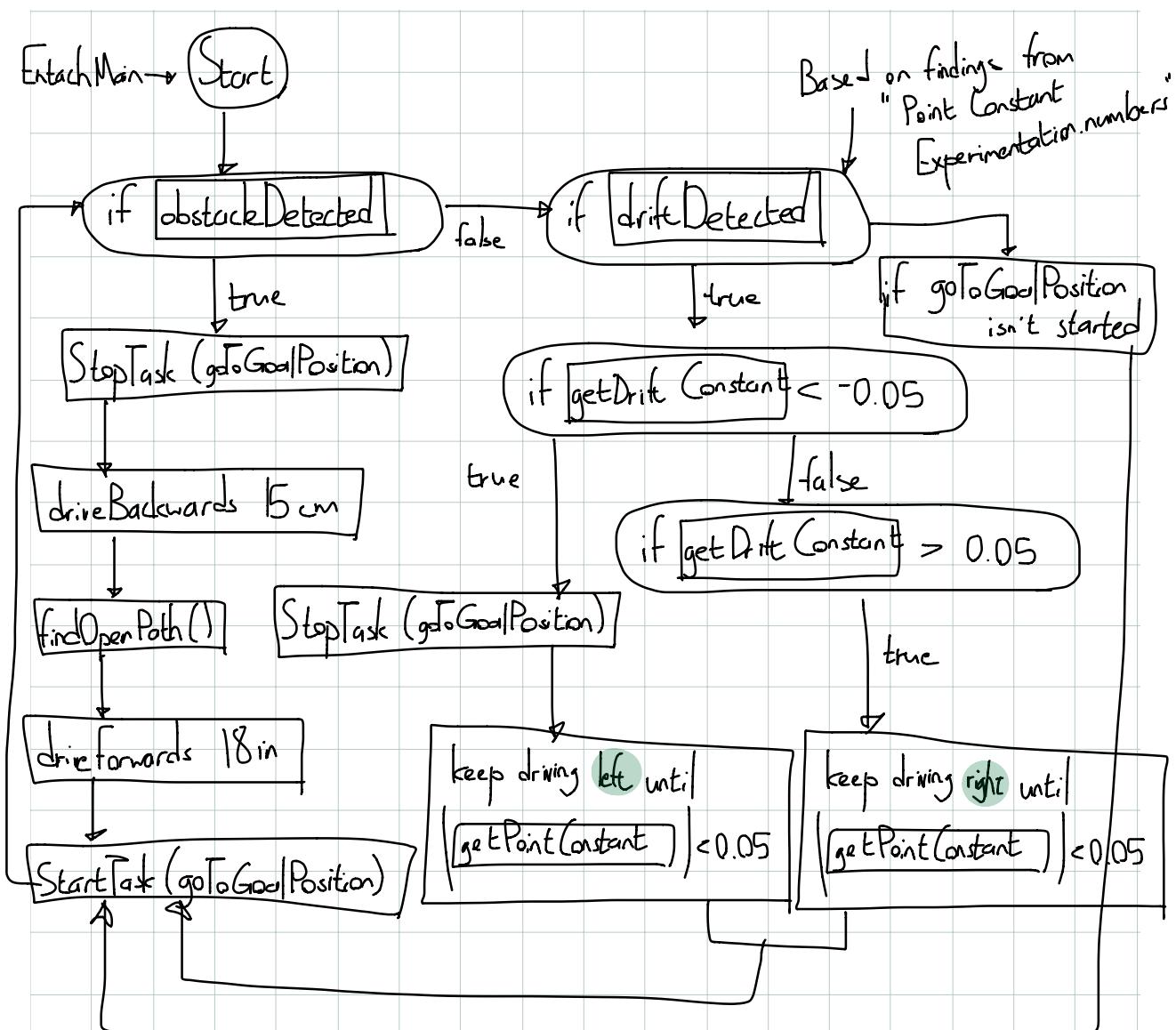
float current X, Y, θ

float lastGoal X, Y

Tasks

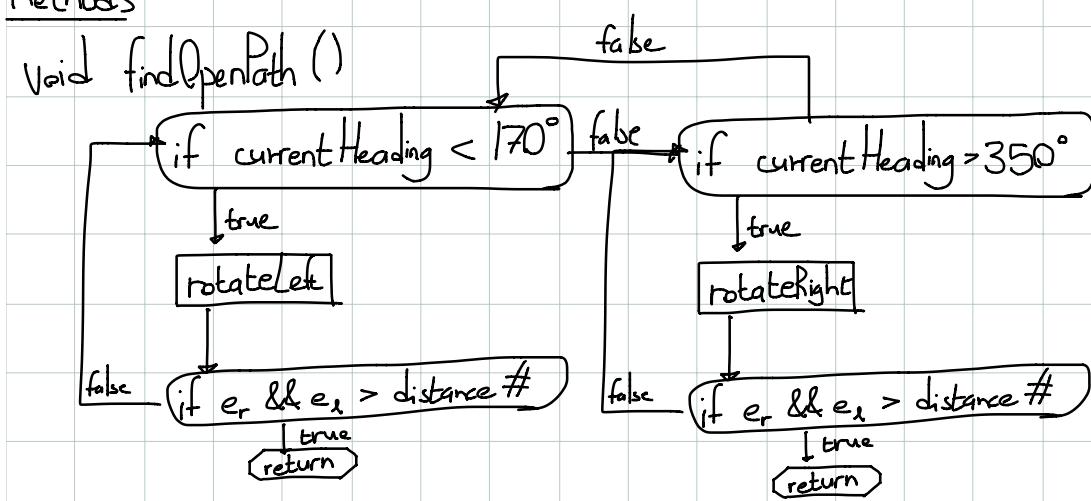
UpdatePosition \rightarrow the 3 equations





Methods

Void findOpenPath()



```
bool detectObstacle()
```

```
    static float speedReading1[3];
```

```
    float speedReading2[3];
```

```
    for each (float reading in speedReading2)
```

```
        reading = encoder[#];
```

```
    bool obstacle Detected
```

```
    if (|speedReading1 - speedReading2| > 5 cm/sec)
```

```
        obstacle Detected = true;
```

```
    else
```

```
        obstacle Detected = false;
```

```
        speedReading1 = speedReading2;
```

```
bool detectDrift()
```

```
if (|getDriftConstant| > 1 cm)
```

```
    return true;
```

```
else
```

```
    return false;
```

```
float getDriftConstant()
```

```
static previousGoal X, Y
```

$$\text{return } \frac{gY_2 - gY_1}{gX_2 - gX_1} (cX - gX_1) - (cY - gY_1)$$