ROBOTICS : 276A HW1 REPORT

PID1: A69034019 Name: Anandhini Rajendran

PID2: A69033245 Name: Yueqi Wu

Video link: https://www.youtube.com/watch?v=AAqyI9ho9gs

Calibration:

MIN SPEED of each motor: vfl: 21 vfr: 21 vbl: 19 vbr: 23

Logic

1. Use `self.theta` to record the current orientation of the robot and update it after every rotation
2. Rotate to the line angle
    a. Line angle is defined as the orientation of the line that starts from the current waypoint to the next waypoint, which is calculated by
    `la = math.atan2(y2-y1, x2-x1)`
    The line angle at each waypoint is shown in table1.
    The line angle is required since the direction of moving to waypoints is not always equal to the difference of orientations.
    b. The angle that needs to be rotated is the difference between current orientation and the line angle, which is `la-self.theta`
    c. Due to the computational error, only do the rotation when
    `abs(la-self.theta) > 0.1`
    d. Rotation speed is constant at 35.
    e. Rotation time is determined by angle change and a rotation constant multiplied since we are keeping the rotation speed constant.
    `abs(int((la-self.theta)*1.59))`
    f. We found the constants and speed by making it rotate a circle and measuring time for a speed.
    g. After rotation, update `self.theta`
3. Move straight in the line directing to the next waypoint
    a. The distance to go is calculated as
    `distance = (((x1-x2)**2+(y1-y2)**2))**0.5`
    b. Fix the time of going straight
    c. Speed is the distance multiplied by a velocity constant. We found the velocity constant by calibrating with multiple inputs.
4. Rotate to the final orientation given by the waypoint

    a. If the current orientation of the robot is different from the orientation given by the waypoint, rotate by the angle difference, then update `self.theta`.
    b. The logic is the same as step 2, but the speed and rotation constant were calibrated separately.
5. Stop for 2 seconds to move to the next point
    a. Useful for the next assignment for taking feedback. Also help us check the status of the robot at each waypoint.

Waypoints	Angle made by line connecting waypoints	Orientation
0, 0	3.14	0
-1, 0	1.57	0
-1, 1	3.14	1.57
-2, 1	1.57	0
-2, 2	-0.78	-1.57
-1, -1	-0.78	-0.78
0, 0		0

Table 1. Coordinates, line angle and orientation at each waypoint

Experiments

1. Rotation
    a. Theta = w * t: Initially, our robot moved with varied speeds and time so we had 2 constants to calibrate for w and t. But at some waypoints the speed was too low for the wheels to move and it's difficult to be calibrated for all kinds of rotations. So we decided to keep one as constant and calibrate the other.
    b. We can keep time constant and calibrate w or keep w constant and calibrate time. Using the first method, w often goes too low or too high and the wheels don't move properly, so we kept w constant and changed the rotation time.
2. Car_mixed
    a. We tried the car_mixed with rotation, translation and sliding by setting four motors as
    ```
    v1 = v_rotate+v_straight+v_slide
    v2 = -v_rotate+v_straight-v_slide
    v3 = v_rotate+v_straight+v_slide
    v4 = -v_rotate+v_straight-v_slide
    ```
    b. We first calibrated it manually, but could not find constants that made sure all four wheels' speeds were within the required range. When `v_rotate`, `v_straight`,

and `v_slide` are mixed together, they cancel each other so it's hard to keep all four wheels within the valid speed range for all kinds of movements.

c. We wrote a program for the linear equation in car mixed to find coefficients for the vx,vy, and w but it resulted in no solution.
d. Hence we went with our initial approach of separating rotation and translation.

Performance

1. Initially, we did only rotation once and then moved straight, but we improved and accounted for the expected orientation of waypoint and move.
2. Since the change in wheel velocity or time can be in units of 1 and not float, we were not able to get the perfect calibration but it was close enough.
3. We drive the robot on the floor, but when it rotates, the wheels sometimes run over sticky notes, which causes the turning angle to be inaccurate and inconsistent every time. Next time, the sticky notes should be placed outside the robot's driving path.
4. It deviates a bit at the end because of the rotation but is unable to reduce since the orientation is too small even if small changes to w or t by 1 unit. If we do this reduction the robot deviates at the middle of the path but goes accurately at the end. So it's a tradeoff on accuracy. The future work would be to use a feedback control loop and take the camera sensor images.

Next Steps

1. Use the kinematic model to set the speed of four motors separately.
2. Measure the parameters of the robot and do the calibration for the above model.