Dean Fletcher

Report – Lab 2

Video: https://youtu.be/LLBELnZaLm0

Task 1:

Given – VL, VR, T

With these known values we're able to travel a target path provided in the lab description. Furthermore, a starting position C pointing north.

Possible moves:

Circle Motion: VL not equal to VR

- this mean we are moving on some type of curve

Straight Motion: VL equal VR and VL is larger than 0

- this means that we are not stopped and moving in a straight line.

Radius Kinematics:

$|\frac{(Axle\ length(\ VL+VR))}{VL-VR}|$ => This result gives Radius based on those velocities

Wheel Velocities:

$PHI = \frac{velocity}{wheel\ radius}$ => This results in the wheels angular speed

PHI should not exceed max of robot at 6.28 radians

Distance:

1. $V = \frac{VL+VR}{2}$
2. $D = \frac{Speed}{Time}$
3. With the Kinematic equation to calculate the linear velocity, and time is known, we can compute distance

Result:

With all these values we are now able to drive the desired x, y, theta

Pose:

- The starting pose is known so with that we can compute our next pose based on the kinematics already discussed.

$$ICC = (x - R * sin(\theta), y + R * \cos(\theta))$$

$$\begin{matrix} x_c \\ y_c \\ \theta_c \end{matrix} = \begin{bmatrix} \cos(\omega t) & -\sin(\omega t) & 0 \\ \sin(\omega t) & \cos(\omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega t \end{bmatrix}$$

- **Straight line**:
  - New theta is equal to last theta since we haven't changed angle (i.e. turned)
  - The new X and Y can be computed by using the current position (x, y) added to the distance traveled and then multiplying by the cos/sin of the angle.
    - new x = x + velocity * time * $cos(\theta)$
    - new y = y + velocity * time * $sin(\theta)$
- **Curve**:
  - At this point we have calculated the radius so now it's time to find our instantaneous center of curvature (ICC).
    - ICC for X = X – R * sin(theta)
      - Denoted as: ICC_x
    - ICC for Y = Y + R * cos(theta)
      - Denoted as: ICC_y
  - We will also need the angular velocity to find the change in angle over the given time.
    - Omega = Velocity / Radius
    - Theta Change = Omega * Time
      - Denoted as: thetaC
  - Finally, all that is left is to compute the pose
    - The new X (or Y) can be broken into 3 parts:
      - X direction = $\cos(thetaC) * (X - ICC_x)$
      - Y direction = $sin(thetaC) * (Y - ICC_y)$
      - Using the already computed ICC in the X direction
      - Combining these results in: $X_{dir} - Y_{dir} + ICC_x$
    - The new Theta is simple computed as:

- New Theta = theta + thetaC

Task 2:

Given – F, C, V

With these known values we're able compute a path like the one provided in the lab description. Furthermore, a starting position C pointing north and final pose F (X, Y, Theta). The assumption is that since there are countless ways to get to target, we follow the procedure from the lab path. This requires that we make two curves (R1, R2) and two straight lines (D1, D2). Secondly, we are assuming that the final pose F is to our right and below us. With this knowledge I began with a PI / 2 (90 Degree) turn. At each step from here we can compute our pose like how it was performed in Task 1. From there we can compute the D1 as the distance left to drive in the X direction. Once we have reached this point, we can take the next curve R2. This curve will compute our current theta in relation to the final theta. This will aid in correcting our orientation. Now for the final distance D2 we compute the distance remaining using the distance formula. With these procedures and the Kinematics discussed in Task 1 we should be able to travel to any final pose that meets the specifications of this assignment.

$$V = wr$$

$$PHI = \frac{V}{r}$$

$$v_l = w(R + \frac{axle}{2})$$

Given the normal Kinematics we can derive a PHI for left and right:

$$PHI_{x,y} = \frac{V - axle * W}{r}, \frac{V + axle * W}{r}$$

Conclusion:

This was a challenging lab where error added up over time and required a lot of tweaking and reevaluating. I learned about how we can use our current orientation and the next desired move to calculate the expected pose after that move. I found that on paper this is accurate but, it can't determine what happens during those motions. So, things like acceleration, slippage, and obstacles are something that will greatly impact the actual pose of the robot over time.