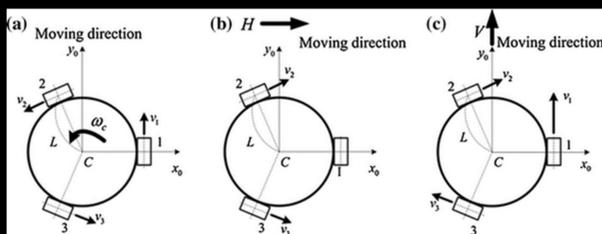


OBJECTIVES

Mobile robots have increased in popularity in recent times due to their ease of mobility and high autonomy. Many new applications take advantage of these features with uses in factories, search and rescue missions, planet exploration, and coordination. In order for these robots to do these tasks, there is a lot of work that goes on behind the scenes. This work examines the task of trajectory tracking for mobile robots. Specifically, omnidirectional mobile robots which use special wheels that have rollers around the circumference of the wheel perpendicular to the rotating axis allowing movement forwards and backwards as well as laterally. Using these wheels we develop a trajectory tracking controller using the kinematic equations of motion for the robot.

APPROACH

In order to develop this trajectory tracking controller, the kinematic equations for the omnidirectional platform of the robot is derived. We then generate a trajectory that we wish for the robot to follow and split it into small segments paired with direction angles. Using these pairs and the kinematic equations, we can calculate the velocities of each wheel as well as the motor encoder count that these wheels should reach before stopping. These velocities and encoder counts are then sent to the robot from MATLAB using WiFi compatible microcontrollers. Onboard the robot we have also developed a Proportional Derivative Integral (PID) controller in order to maintain velocities for each wheel and stop the motion of the wheel when the desired encoder counts are met.



CONTACT INFORMATION

Email: imngui@uwm.edu

Simulation Videos

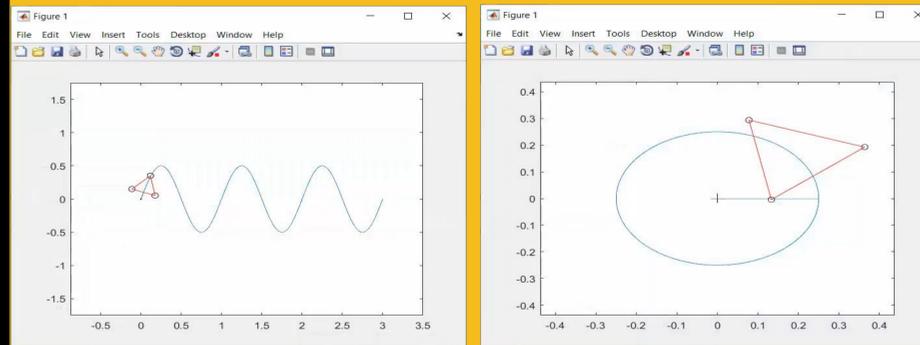


Fig. 1

Fig. 2

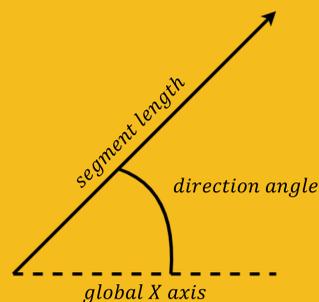
METHODOLOGY

Kinematic Equations

$$\begin{aligned}\dot{\varphi}_1 &= (-\sin(\theta) \cos(\theta) \dot{x}_L + \cos^2(\theta) \dot{y}_L + R\dot{\theta})/r \\ \dot{\varphi}_2 &= (-\sin(\theta + \alpha_2) \cos(\theta) \dot{x}_L + \cos(\theta + \alpha_2) \cos(\theta) \dot{y}_L + R\dot{\theta})/r \\ \dot{\varphi}_3 &= (-\sin(\theta + \alpha_3) \cos(\theta) \dot{x}_L + \cos(\theta + \alpha_3) \cos(\theta) \dot{y}_L + R\dot{\theta})/r\end{aligned}$$

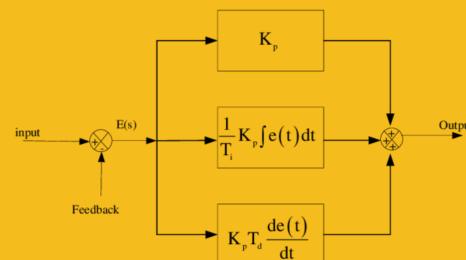
$\dot{\varphi}_i$ is the angular velocity of wheel i
 α_i is the offset angle of each wheel from the origin
 $\theta, \dot{\theta}$ are the orientation and the rotational velocity
 R, r are the radius from the center of the robot to the wheel and radius of the wheel
 \dot{x}_L, \dot{y}_L are the local x and y directions

Breaking down trajectory into segment lengths and direction angles



PID Controller

PID controllers use past and present error to maintain a set value. The motors used on our robot have encoders in them that tell us how many ticks of the encoder occurred in a certain amount of time. Since we know the amount of ticks in one rotation we can determine the velocity of the wheel. We pass in velocities for each wheel to the PID controller and it maintains those velocities as well as stops the motor when it reaches a specific encoder count as that wheel has reached its end point.



RESULTS

Simulations were performed using the kinematics and PID controller. We ran simulations on a sine curve trajectory and on a circular trajectory. Fig. 1 shows a video of the simulations results on the sine curve trajectory. As you can see in the video, the center of the robot always stays on the trajectory we wish to follow.

In Fig.2 we simulate the robot following a circular trajectory. From the simulation we can also see that the controller is capable of following trajectories like this.

This is helpful in simulating how a robot will move around corners or along curved paths which tend to be more difficult for traditional robots due to the lack of the omnidirectional property.

CONCLUSION

This research shows that our kinematics are working properly and from the simulations we should get good experimental results. It also shows that our method of breaking down the trajectory works properly. In future research, we plan to perform the physical tests utilizing the PID controllers that are already setup and running in under 1ms. We also plan to add a robotic arm on top of the platform and use those robots for coordination.

BIBLIOGRAPHY

- Bui T.L. (2016) Decentralized Motion Control for Omnidirectional Mobile Platform—Tracking a Trajectory Using PD Fuzzy Controller. In: Duy V., Dao T., Zelinka I., Choi H.S., Chadli M. (eds) AETA 2015: Recent Advances in Electrical Engineering and Related Sciences. Lecture Notes in Electrical Engineering, vol 371. Springer, Cham
- Optimization pitch angle controller of rocket system using improved differential evolution algorithm - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Block-diagram-of-PID-controller-10-14_fig2_318695712 [accessed 17 Apr, 2020]