

Path tracking mobile robot using steppers

Ankit Deo^{1,*}, Ayush Gupta², Himanshu Khemani¹ and Rashmi Ranjan Das³

¹School of Electrical Engineering Vellore Institute of Technology, Vellore

²School of Mechanical Engineering Vellore Institute of Technology, Vellore

³Associate professor Vellore Institute of Technology

Abstract. In control of mobile robots, precision plays a key role in path tracking. In this paper we have intended to use hybrid stepper motors for precise control of the two wheeled robot. A control algorithm was developed to control the robot along different trajectories. We have found that stepper motors are more accurate for path tracking than normal DC motors with wheel encoders and one can obtain the implicit coordinates of the robot in runtime more precisely. Getting the precise coordinates of the robot at runtime can be used in various SLAM and VSLAM techniques for more accurate 3D mapping of the environment.

1 Introduction

Robotics has become an indispensable part in our lives from strategic military operations to mundane applications of household and cleaning. Among the various applications of robotics path tracking have been one of the most emerging fields in terms of research. Legged mechanism can be used for most precise mapping of the trajectory. However controlling and operating legged mechanism robots are much more complicated than wheeled robots. Apart from the control systems legged mechanism robots are relatively slower. Differential drive odometry techniques with DC motors and encoders are used for path tracking algorithms. Wheel encoder output gives velocity in which the two wheels move. The bot can be moved along a certain curve by giving the two wheels different velocities. Such a bot will have problems moving along a straight line since both the wheels are independent. Moving both wheels with same velocity is a challenge. So our research is based on a better approach of using stepper motors rather than DC for precise rotation of the wheels and in turn the orientation of the bot. To keep track of orientation of the bot at a particular point a magnetometer is also used. The error caused in the actual orientation of the bot given by magnetometer and the orientation achieved due to movement of stepper motor is also taken in consideration. In traditional differential drive odometry readings from the encoders are sent as a feedback to the system to determine the motion of the wheels and the position of the bot. In our stepper model of the robot stepper motors are used for the precise motion of the bot. The wheels can be independently controlled to change the orientation of the bot at any given instant of time.

2 Types of stepper motors

Based on construction and working there are three types of stepper motor –

2.1. Variable reluctance stepper motor

Soft iron multi tooth rotor and a wound stator is used in the variable reluctance stepper motor[3]. The rotor rotates at a specific angle which minimizes the reluctance between opposite windings in the stator of the stepper motor. This type of stepper motor offers supreme level angular resolution but has low torque.

2.2 Permanent magnet stepper motor

Permanent magnets are added to the motor's structure instead of teeth. The rotor is magnetized with alternating north and south poles aligned along the straight lines parallel to the rotor shaft.[4] Permanent magnet stepper motor has better torque characteristics due to increased magnetic flux density provided by the magnetized rotor poles[4]. It has low cost and low resolution as compared to other types. It is an incremental device which responds to discrete commands by rotating the output shaft in small equal steps for each of the discrete command.

2.3 Hybrid stepper motors

Features of both Variable reluctance motor and permanent motor are included in a hybrid stepper motor for better efficiency.[7] It has very good angular resolution, speed and torque. Hybrid motor contains a multi tooth rotor which has axially magnetized concentric magnet surrounding its shaft. The magnetic flux is guided to preferred locations in the air gap by the path provided by the teeth on the rotor. This increases the detent and holding torque properties of the motor when compared with other two types of stepper motor.

* Corresponding author: ankitdeo7@gmail.com

2.3 Stepper vs DC

Normal Dc motors have continuous output whereas stepper motors have step outputs. [2]For precise intrinsic coordinate mapping of our two wheeled mobile robot we found that stepper motors are indeed better than normal dc motors equipped with wheel encoders which give rotations as a feedback in terms of ticks. Which acts as a closed loop control system and renders instability to the overall system.

3 Solidworks modelling

Traditional differential drive wheeled robots are maneuvered with the help of DC motors. Standard chassis are readily available for these kinds of bots. We have designed our own model of chassis as chassis incorporating stepper motors are not designed previously. [1]Our design is a simple three wheel configuration in which the rear wheels are controlled using stepper motors and the front wheel is a castor wheel. This configuration helps for smooth point rotation and lateral movement of the bot. For our research we made solidworks model of each part i.e. wheels, motors, chassis, battery, etc. separately and then joined them in as assembly model in 1:1 scale. The weight and dimensions for each part is manually entered so that the centre of gravity of the assembly is automatically calculated by solidworks . The position of centre of gravity is critical because the magnetometer is placed in orientation with the centre of gravity to measure the orientation of the robot. The assembly model of the robot was saved in xml file format to import into MATLAB to run simulation.

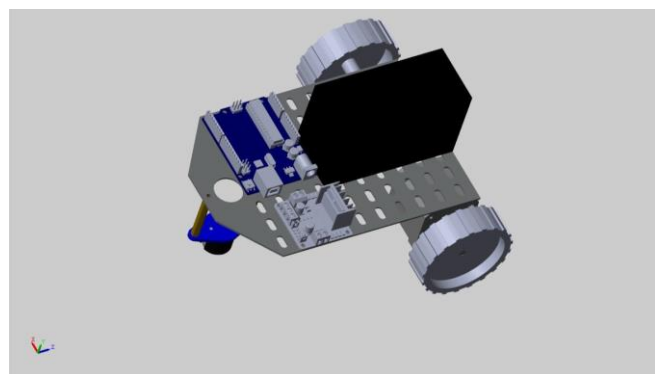
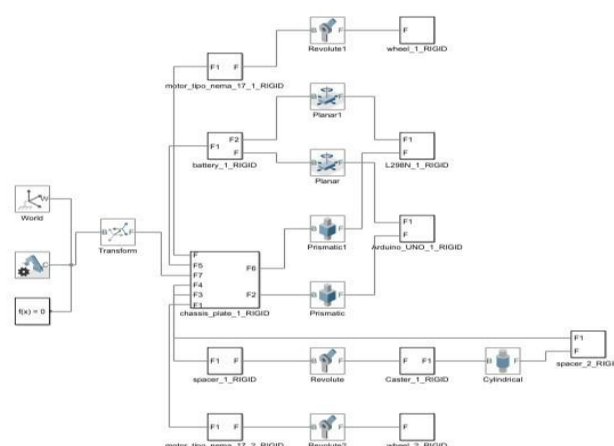


Fig. 1. Solidworks model with the exact specifications exported to simulink.

4 Solidworks model imported in MATLAB and SIMULINK.

Sim mechanics a library in MATLAB Simulink environment is used to import any kind of solidworks model into Simulink environment. Our bot with three revolute [5]joints two at the rear attached with the stepper and one at the front attached with the castor wheel is imported in the matlab environment fig. Parameters of both the wheels were varied to observe the behavior of the entire model in Simulink environment

Fig. 2. Simulink model



which is shown in figure 2. Following results were obtained after performing the simulation. For our research we used a stepper motor with 1.8 degrees of rotation in a step. For path planning algorithms we need to know the coordinates of the bot at every given frame. The linear displacement of the bot can be calculated by measuring the angular displacements of the stepper motor shaft. At every frame the bot will have some change in its orientation. This change is measured from a magnetometer. The readings obtained from the

magnetometer acts as an input for our rotation matrices. Subsequent rotation matrices are multiplied in MATLAB Simulink model to get orientations at any given frame of reference.[8] The rotation matrix along with the given displacement factor and scaling factor acts as the input for our homogeneous transformation matrix. We have multiplied the homogeneous matrices in series just like rotation matrices in series in MATLAB Simulink environment. The implicit coordinates of the bot can be calculated at any given point through this.

5 Trajectory Mapping

Every trajectory can be expressed in terms of mathematical function. In our research we have considered continuous and differentiable trajectory functions. Trajectories expressed in terms of $y=f(x)$ are sampled along the x axis. Continuous curved trajectories can be sampled via simple commands in MATLAB.Our simulation shows an example of a sine wave sampled along the x axis. At every sampled point x we have found the corresponding value of $f(x)$ in our case $\sin(x)$.for very two consecutive points X_n and X_{n-1} , we get the corresponding $f(X_n)$ and $f(X_{n-1})$ values. Using coordinates $f(X_n)$, $f(X_{n-1})$, X_n , X_{n-1} and the distance formula, we can obtain the linear distance the bot needs to move between two subsequent points in order to follow the desired trajectory. Distance d is calculated using the equation 1.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

Using these four coordinates we can determine the change in orientation the bot needs to go through, to reach the adjacent point. In our research we have calculated the change in orientation the bot needs to go through between every two adjacent points, in order to reach the next adjacent point.

$$\text{steps per revolution of stepper} = 360^\circ/1.8^\circ \quad (2)$$

At every step the bot is programmed to change its orientation, Instead of changing it's orientation only at the sampled points. This helps to minimise the error in trajectory mapping since the curve traced after subsequent changes in orientation is relatively smoother.[6] For changes in orientation angles lesser than 1.8 degrees we have used half stepping and 1/8th stepping in stepper motors. 1/2 stepping and 1/8th stepping in stepper motors: Implementation of half stepping and 1/8th stepping helps in precise motion of our bot, since stepping action is not limited to 1.8 degrees. The error is plotted in table 1. For movements that require change orientation between points lesser than 1.8 degrees we have implemented 1/2 stepping and 1/8th stepping. Illustrated using an example where we considered relatively smoother trajectory in comparison to the sine wave. $F(x) = x^2$. This is shown in figure 3.

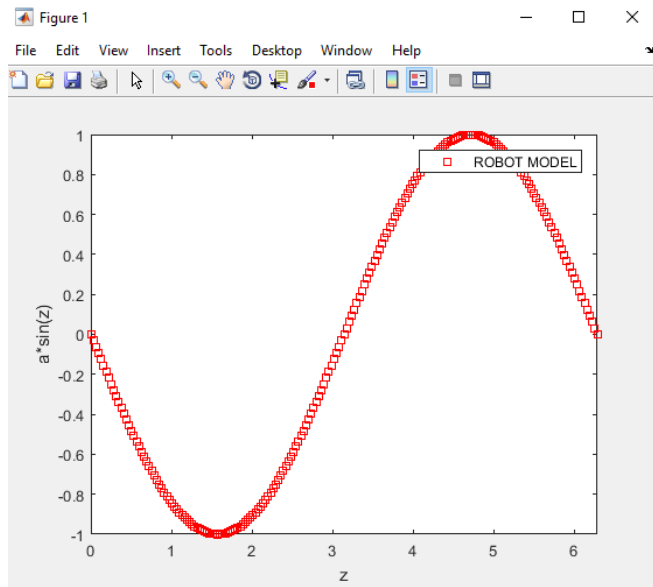


Fig. 3. Path generation (plotted in MATLAB).

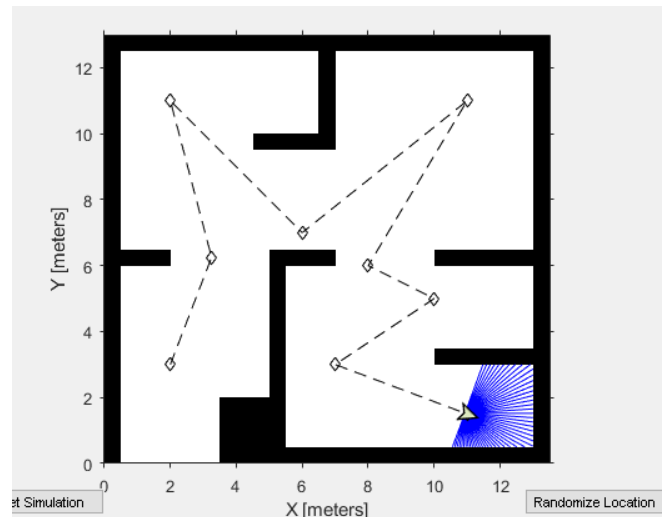


Fig. 4. Path generation using LIDAR simulation. (in MATLAB).

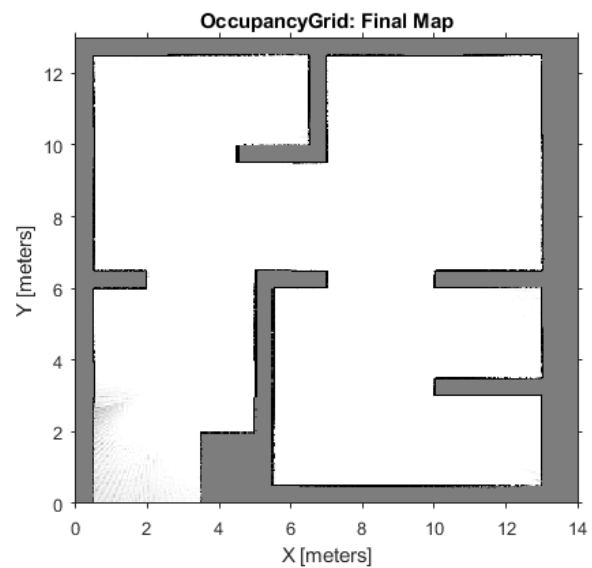


Fig. 5. Binary occupancy grid generated by LIDAR simulation.

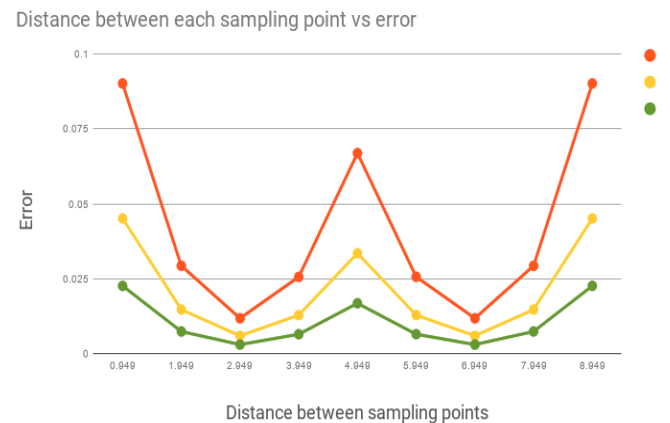


Fig. 6. Error plot for different stepping of stepper motors ..

Table 1. Data for path tracking

| z(in m) | theta (rad) | theta (deg) | Theta/1.8 | no of rotations | no of steps for stepper motor | no of steps for stepper motor (1/2 stepping) | no of steps for stepper motor (1/4 stepping) | error | for full stepping (cm) | for full stepping (1/2 stepping)(cm) | for full stepping(1/4 stepping)(cm) |
|---------|-------------|-------------|-----------|-----------------|-------------------------------|--|--|-------|------------------------|--------------------------------------|-------------------------------------|
| 0.949 | 0.7441 | 42.655 | 23.69745 | 4.46979 | 893.958 | 1787.916 | 3575.832 | 0.958 | 0.090243 | 0.045126 | 0.02256 |
| 1.949 | 0.4555 | 26.111 | 14.50636 | 9.17979 | 1835.95 | 3671.916 | 7343.832 | 0.310 | 0.029277 | 0.014636 | 0.00731 |
| 2.949 | 0.168 | 9.659 | 5.366242 | 13.8897 | 2777.95 | 5555.916 | 11111.832 | 0.124 | 0.011718 | 0.005858 | 0.00292 |
| 3.949 | 0.643 | 36.91 | 20.50636 | 18.5997 | 3719.95 | 7439.916 | 14879.832 | 0.271 | 0.025565 | 0.012788 | 0.00639 |
| 4.949 | 0.775 | 44.43 | 24.68789 | 23.3097 | 4661.95 | 9323.916 | 18647.832 | 0.710 | 0.066957 | 0.033476 | 0.01673 |
| 5.949 | 0.643 | 36.91 | 20.50636 | 28.0197 | 5603.95 | 11207.916 | 22415.832 | 0.271 | 0.025603 | 0.012806 | 0.00640 |
| 6.949 | 0.168 | 9.659 | 5.366242 | 32.7297 | 6545.95 | 13091.916 | 26183.832 | 0.124 | 0.011718 | 0.005858 | 0.00292 |
| 7.949 | 0.4555 | 26.117 | 14.50636 | 37.4397 | 7487.95 | 14975.916 | 29951.832 | 0.310 | 0.029277 | 0.014636 | 0.00731 |

Conclusion

In this research we were able to successfully traverse a sine wave trajectory with 10 points of sampling and 100 points of sampling. Errors observed were much lesser which is plotted in figure 6. than what we observed for traditional differential drive robots with normal DC motor. For angular resolutions lesser than 1.8 degrees concept of half stepping, micro stepping and one eighth stepping can be used. Using stepper motors made the system more energy efficient and stable as it is an open loop control system. Using the coordinates calculated after multiplying rotation matrices we used them for VSLAM. A LIDAR simulation in MATLAB shown in figure 4. We obtained a binary occupancy grid showing free path and obstacles as shown in figure 5.

References

1. Ilya K and N.Harris M . Developments in nonholonomic control problems. *IEEE Control Systems* 1995; 15(6):20 – 36.
2. Lydia EK, Petr S, Jean CL and Mark HO. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robotics and Automation* 1996; 12(4):566–580.
3. Jean PL, Paul EJ, Michel T, Richard and Murray M. A motion planner for nonholonomic mobile robots. *IEEE Trans. Robotics and Automation* 1994; 10:577–593.
4. Ankit Deo, Raaghavi Ravisankar. Differential Drive odometry for two wheeled mobile robot. *IJRSET*.
5. Trajectory tracking and control of differential drive robot for predefined regular geometrical path Global Colloquium in Recent Advancement and Effectual Researches in Engineering, Science and Technology (RAEREST 2016)

8. Kuniaki K, Liang M, Jianru X, Chengwei Z and Nanning Z. A path generation for automated vehicle based on Bezier curve and via-points. *Robotics and Autonomous System*. 2015; 74:243–252. [6] Steven ML. Rapidly-exploring random trees: A new tool for path planning. Tech. Rep. Computer Science Dept. Iowa State University 1998.
9. James JK and Steven ML. RRT-Connect: An Efficient Approach to Single-Query Path Planning, *ICRA 2000*; 1-7. [8] Ákos N, Gábor C and Domokos K. Path Planning and Control of Differential and Car-like Robots in Narrow Environments. *SAMI 2015*;103-108.
10. Bong SP, Sung JY, Jin BP, and Yoon H C. A Simple Adaptive Control Approach for Trajectory Tracking of Electrically Driven Nonholonomic Mobile Robots. *IEEE Trans. Control Syst. Technol* .2010; 18(5):1199-1206.
11. Elie M, Maarouf S and Hamadou S. A higher level path tracking controller for a four-wheel differentially steered mobile robot. *Robotics and Autonomous Syst*. 2006; 54:23–33.