Energy-Efficient Path Planning and Obstacle Avoidance for Differential Wheel Robots Using Model Predictive Control

Sunday Oyijewu Ochedi

A20503423

Abstract-**This article discusses the development and implementation of a path planning and obstacle avoidance strategy for differential wheel robots. Utilizing the A* algorithm for path-planning and Model Predictive Control (MPC) for trajectory tracking and energy optimization. This project aims to enhance robotic navigation in complex environments. The primary focus is on minimizing energy consumption while ensuring effective obstacle navigation and precise path adherence.**

Keywords: A* algorithm, Model Predictive Control, Path Planning, Obstacle Avoidance, Energy Efficiency.

I. INTRODUCTION

The increasing demand for autonomous mobile robots in industrial, domestic, and military applications calls for advanced path planning and obstacle avoidance techniques that optimize both performance and energy efficiency. This project integrates the A* algorithm with Model Predictive Control (MPC) to address these needs, ensuring the robot can navigate efficiently through environments with obstacles.

II. PROJECT OBJECTIVES AND OVERVIEW

The project's main objectives are:

Path Planning: Employ the A* algorithm to determine the shortest path between designated start and goal points while avoiding obstacles.

Energy Efficiency: Implement MPC to follow the path determined by the A* algorithm and optimize the control inputs to minimize energy usage, quantified through the kinetic energy of the robot.

III. RELATED WORK

The A* algorithm was first proposed in [2] and is one of the best path-planning algorithms. The algorithm is a heuristic search algorithm that is used to find the shortest path based on information of obstacles present in a static environment [4]. The algorithm does this by first selecting feasible node pairs and then calculating the shortest route based on these pairs [1].

Model Predictive Control (MPC) is valued for its ability to manage complex, constrained systems with multiple inputs and outputs. It calculates an optimal sequence of controls over a set period, applying only the first input to optimize system performance continuously [3]. Since only finite horizon problems are solved in each MPC time step, closed-loop stability may not hold [6]. However, imposing terminal constraints may help ensure stability [5].

IV. METHODOLOGY

A. A* Algorithm

The A* algorithm provides a foundation for path planning by calculating the least cost path from start to goal, factoring in both the exact path cost and an estimated cost to the goal. The cost function is defined by:

$$F(n) = G(n) + H(n)$$

$G(n)$ represents the exact cost from the start point to any node n. While $H(n)$ is the heuristic function that estimates the cost of the cheapest path from n to the goal. $H(n)$ is defined in this case as the Euclidean distance defined as:

$$H(n) = \sqrt{\left(x_{\mathrm{n}} - x_{goal}\right)^2 + \left(y_{\mathrm{n}} - y_{\mathrm{goal}}\right)^2}$$

B. Model Predictive Control (MPC)

MPC is used to refine the robot's trajectory along the A* planned path. The control model considers the robot's kinematics and implements constraints on velocity and state variables to maintain the planned course and manage obstacle avoidance effectively. The purpose of the MPC control action is to minimize this cost function:

$$J = \sum_{k=0}^{N-1} \left(x(k) - x_{ref}(k)\right)^T Q \left(x(k) - x_{ref}(k)\right) + \left(x(k) - x_{ref}(k)\right)^T R \left(x(k) - x_{ref}(k)\right)$$

Where J is the operating cost, N is the prediction horizon and matrices Q and R, are $(n \times n)$ and $(m \times m)$ respectively, they are symmetric positive definite weight matrices.

C. System Model and Constraints

The robot's kinematic model includes:

$$\dot{x}(t) = f_c\big(x(t), u(t)\big)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v cos\theta \\ v \sin\theta \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

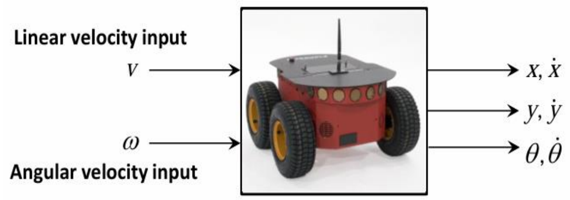

Fig.1: Input, output, and states of a differential mobile robot

Euler Discretization

$$x(k+1) = f\big(x(k), u(k)\big)$$

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} + \Delta T \begin{bmatrix} v(k)\cos\theta(k) \\ v(k)\sin\theta(k) \\ w(k) \end{bmatrix}$$

Kinetic energy (KE):

One objective of this project is to minimize the energy used by the robot, defined by the kinetic energy:

$$kE = {}^1/_2\, mv^2$$

For simplicity, the mass of the robot is assumed to be 2kg. This leaves us with a simplified equation:

$$kE = v^2$$

By minimizing velocity v, MPC directly reduces kinetic energy.

Linear and Angular Velocities: Constrained by maximum values to ensure safe and practical operation speeds.

$-v_{max} \leq v_{max}$

$-w_{max} \leq w_{max}$

Where:

$v_{max} \;=\; 2\ \mathrm{m/s}$

$$w_{max} \;=\; \mathrm{pi}/4\ \mathrm{rad/s}$$

State Constraints: Enforce boundaries on the robot's position and orientation, facilitating operation within a predefined area.

$$x_{min} \leq x \leq x_{max}$$

$$y_{min} \leq y \leq y_{max}$$

$$\theta_{min} \leq \theta \leq \theta_{max}$$

Where:

$$x_{min} = 0,\ x_{max} = 50 \text{ m}$$

$$y_{min} = 0,\ y_{max} = 50 \text{ m}$$

$$\theta_{min} = -\pi,\ \theta_{max} = \pi$$

Obstacle avoidance is managed through geometric constraints, ensuring the robot maintains a safe distance from any potential collisions.

$$\sqrt{(x - x_0)^2 + (y - y_0)^2} \geq r + r_0$$

The function below is set up as the obstacle constraint to help change the robot's path when it encounters an obstacle.

g = [g; -sqrt((P(4)-obs_x)^2+(P(5)-obs_y)^2) + (obs_diam/2 + obs_diam/2)];

Where:

P(4)= x_{ref},   P(5)= y_{ref}

CASADI toolkit: The CASADI tool is an open-source nonlinear MPC tool used to simulate this control problem.

V. RESULTS AND DISCUSSION

Initial simulations using the A* algorithm helped the robot to plot a short path between its start point and goal destination while avoiding obstacles.

Fig.2 below shows the generated path.

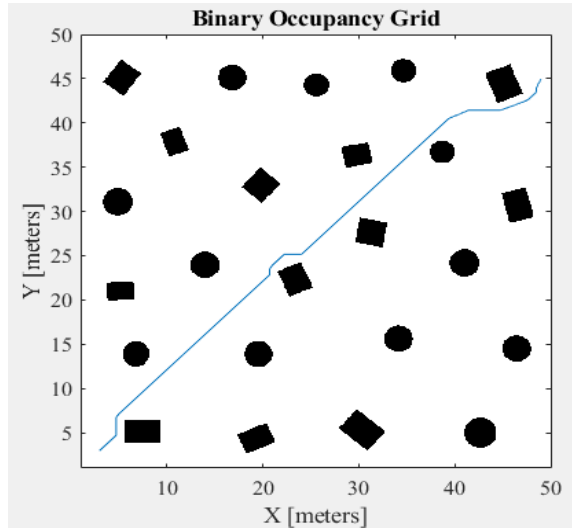

Fig.2: Path planned by A* algorithm.

This path was then used with the MPC controller to drive the mobile robot to the goal point by following the planned trajectory. This is shown by the red line plot in Fig.3 and Fig.4. Fig.3 shows the initial position of the robot. While Fig.4 shows its trajectory to the goal point.

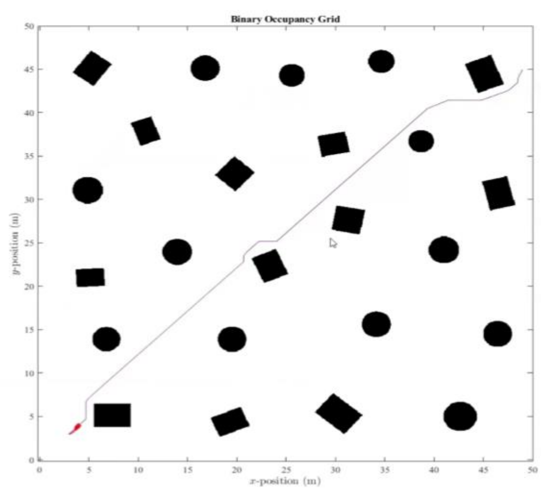

Fig.3: Robot's initial position

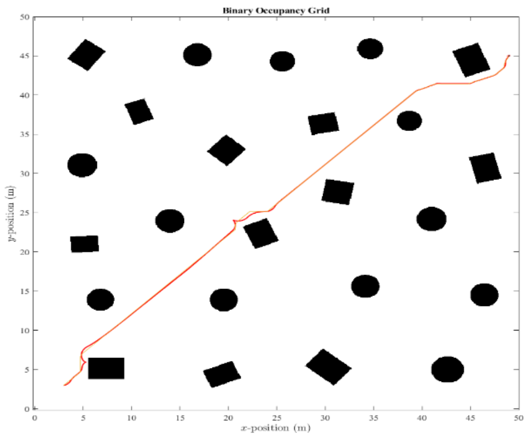

Fig.4: MPC trajectory tracking

This demonstrated the robot's capability to identify paths. But this required further refinement to handle dynamic obstacles. To achieve this, a random obstacle was placed in the robot's path to evaluate its adaptive navigation capabilities. This obstacle was placed after the A* algorithm had generated a path for the robot to ensure that the A* algorithm did not account for the new obstacle. Fig. 5 shows the obstacle which is a blue circle in the robot's path.

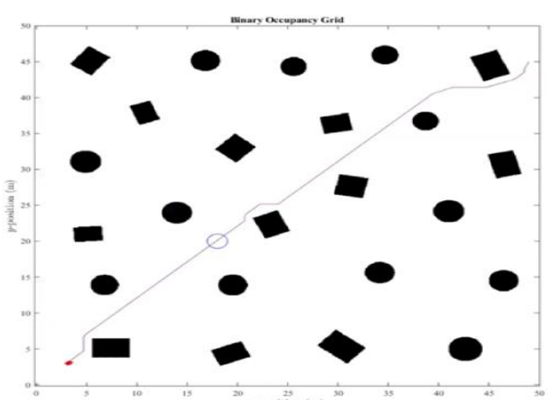

Fig.5: New obstacle to evaluate the robot's adaptive navigation capabilities.

Incorporating the obstacle constraints in MPC significantly enhanced path adherence and obstacle navigation. This helped the robot to navigate around the obstacle and return to its reference trajectory as shown in Fig.6.

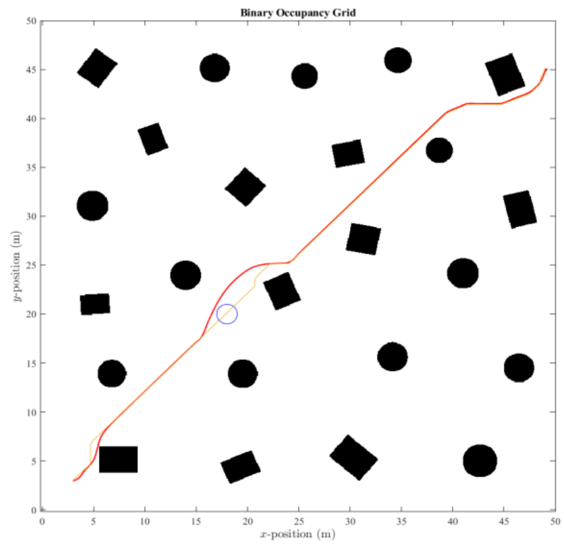

Fig.6: MPC obstacle avoidance and trajectory tracking.

Control plots and error analyses confirm the effectiveness of the implemented strategies, particularly in minimizing energy consumption through optimized velocity control.

Fig.7,8,9, and 10 shows the robot's linear and angular velocities as well as the energy used and X and Y errors while tracking the reference trajectory using the MPC controller.

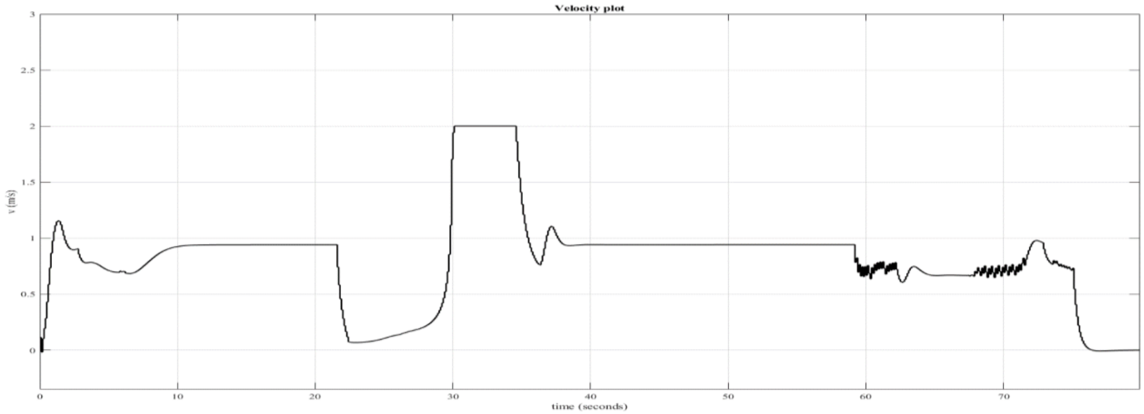

Fig.7: Robot Linear velocities.

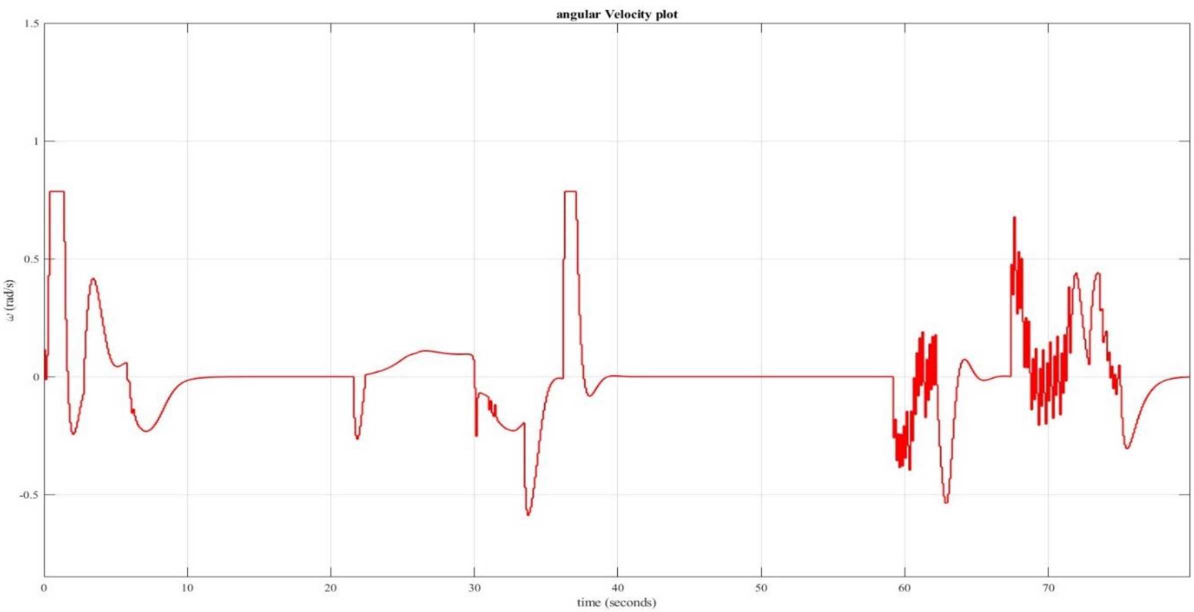

Fig.8: Robot’s angular velocities.

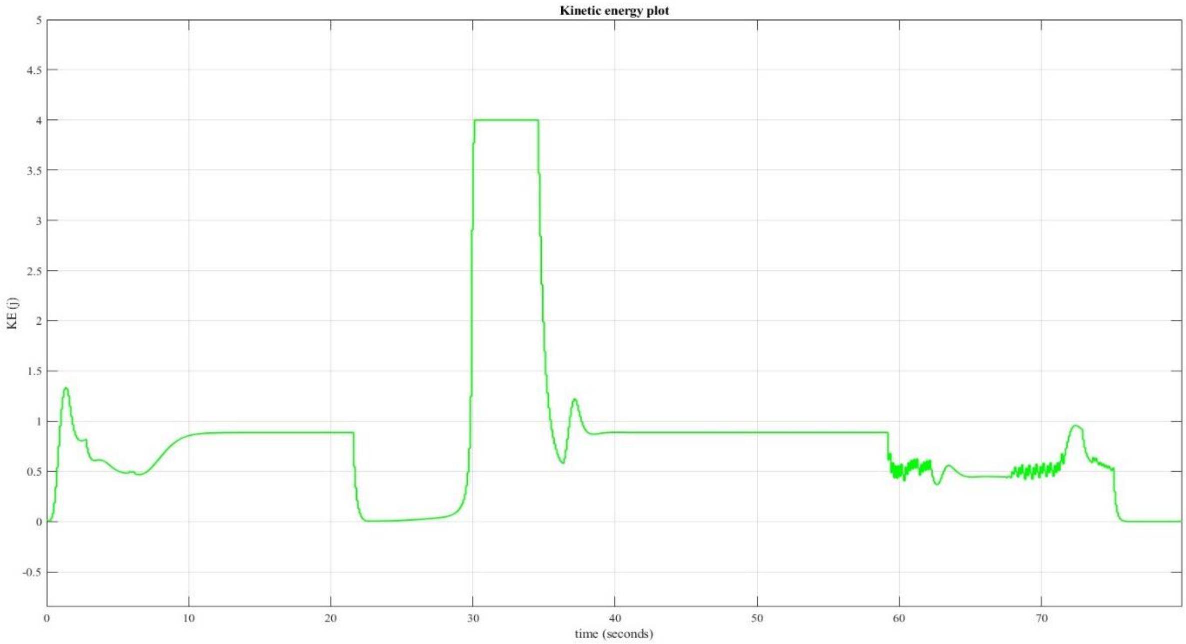

Fig.9: Robot’s energy used.

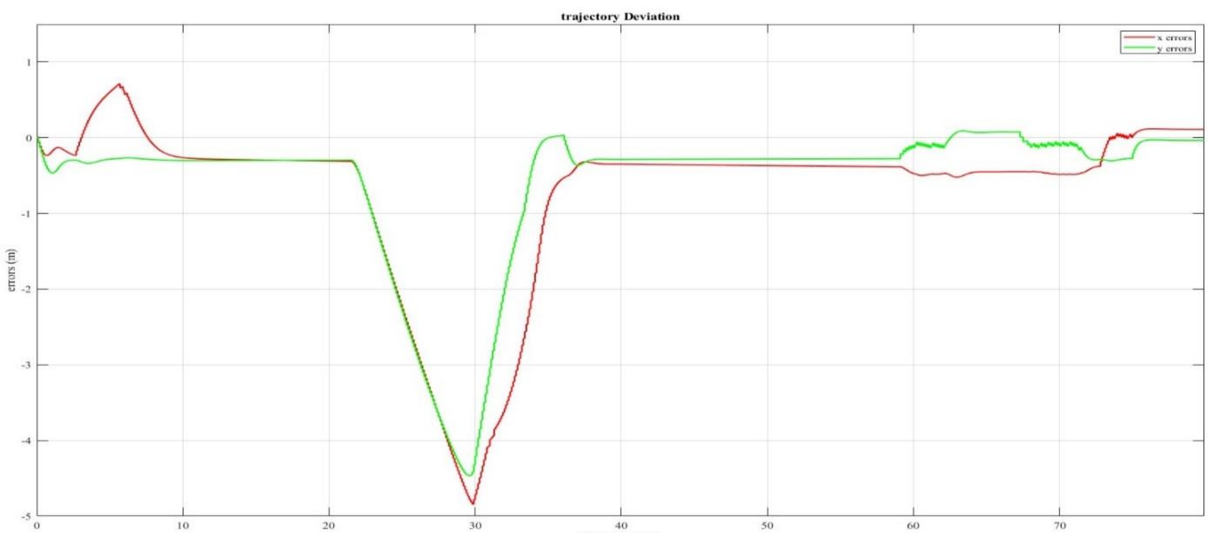

Fig.10: Robot’s X and Y trajectory error.

VI. CONCLUSION

The integration of A* pathfinding with MPC has proven effective in improving both the energy efficiency and operational accuracy of differential wheel robots in obstacle-ridden environments. Future work will focus on real-time adaptation to dynamically changing environments and further optimization of energy usage.

REFERENCES

[1] T. Dudi, R. Singhal, and R. Kumar, "Shortest Path Evaluation with Enhanced Linear Graph and Dijkstra Algorithm," in Proc. 59th Annu. Conf. Soc. Instrument and Control Eng. Japan (SICE), 2020, pp. 451–456.

[2] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE Trans. Syst. Sci. Cybern., vol. 4, no. 2, pp. 100–107, 1968.

[3] J. B. Rawlings and D. Q. Mayne, Model Predictive Control: Theory and Design. Madison, WI: Nob Hill Publishing, 2009.

[4] K. Karur, N. Sharma, C. Dharmatti, and J. E. Siegel, "A survey of path planning algorithms for mobile robots," Vehicles, vol. 3, no. 3, pp. 448–468, 2021.

[5] S. Keerthi and E. Gilbert, "Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations," J. Optim. Theory Appl., vol. 57, no. 2, pp. 265–293, 1988.

[6] T. Raff, S. Huber, Z. K. Nagy, and F. Allgöwer, "Nonlinear model predictive control of a four tank system: An experimental stability study," in Proc. IEEE Conf. Control Appl., 2006, pp. 237–242.