ITMO University
(ITMO)

International Research and Educational Center for
Physics of Nanostructures

SYNOPSIS
for the
subject
‘Robot Programming’

on the topic:
TriNav - Tri-Layer Navigation (A*, DWA, MPC)

Student:
Group No. 478389 *Ayan, Ali*
Group No. 472268 *Umer, Ahmed Baig*
Group No. 475214 *Urwa, Mughal*

Tutor:
Professor *Artemov Kirill*

Saint Petersburg 2025

Contents

Introduction

Unmanned Ground Vehicles (UGVs) play a pivotal role in autonomous navigation, particularly in applications such as logistics, surveillance, urban mobility, and environmental monitoring. Achieving real-time, robust, and safe navigation in unstructured or semi-structured environments remains a key challenge, especially for small to mid-scale robots with differential drive configurations. These systems must cope with static and dynamic obstacles, maintain smooth control, and operate within the constraints of their limited sensing and actuation capabilities.

Modern robotics research emphasizes hybrid planning and control architecture that blend high-level path planning with reactive and predictive local control. State-of-the-art solutions combine discrete global path planners like A* with continuous-time optimization techniques like Model Predictive Control - MPC and reactive local planners like Dynamic Window Approach - DWA to enhance both safety and adaptability. However, integrating these layers efficiently, especially on resource-constrained UGVs, requires careful synchronization of trajectory generation, obstacle avoidance, and control execution.

1.1 Main Goal and Objectives

Design and implement a robust, real-time hierarchical navigation system for a differential-drive UGV that ensures safe, smooth, and efficient path execution in static cluttered environments, using a modular architecture of A*, DWA, MPC, and PID control.

1. Specific Objectives:
   - Global Path Planning: Compute an obstacle-free path from start to goal using A*.
   - Reactive Local Planning: Use DWA to generate collision-aware velocity commands.
   - Trajectory Optimization: Refine the DWA output via short-horizon MPC for smoother execution and obstacle avoidance.
   - Precise Actuation: Convert optimized commands into motor inputs via PID control.
   - Performance Visualization: Provide plots and metrics (speed, path, heading) to evaluate execution efficiency.

1.2 Motivation

Non-holonomic constraints in differential-drive systems challenge traditional control methods. A layered control approach balances completeness (via A*), safety (via DWA), optimality (via MPC), and hardware feasibility (via PID).

Combining DWA and MPC leverages DWA's fast local collision handling and MPC's optimization capabilities, aligning with hybrid designs demonstrated in state-of-the-art literature.

1.3 Problem Statement

The core problem addressed in this project is:

How can a differential-drive UGV autonomously and reliably navigate from a start to a goal position in a cluttered static environment using a modular architecture that combines global planning, local reactive behavior, and control optimization under real-time constraints?

1. Recent literature highlights limitations in individual planning or control approaches:
   - Global planners like A* ensure completeness but lack adaptability to real-time hazards.
   - Local planners like DWA are reactive but can get trapped in local minima or generate suboptimal paths.
   - Model Predictive Control (MPC) offers trajectory optimization but requires robust feedforward references and is computationally intensive.
2. To address these gaps, we propose a hybrid navigation stack consisting of:
   - A* for not smooth global path planning using an obstacle-based grid map,
   - DWA for short-term collision-aware trajectory prediction and selection,
   - MPC to refine local trajectories by optimizing smoothness, obstacle cost, and control effort,
   - PID control to accurately execute the desired velocity commands on a differential drive system.

This layered strategy enables the robot to plan efficiently, adapt locally to obstacles, optimize its behavior dynamically, and remain stable and responsive under actuation constraints.

2 Dynamic Modeling

2.1 Non-holonomic Kinematics

The base continuous model is:

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \omega$$

Discretized with first-order filters:

$$\begin{aligned}
v_{t+1} &= v_t + \frac{v_{\text{cmd}} - v_t}{\tau_v}\Delta t, \\
\omega_{t+1} &= \omega_t + \frac{\omega_{\text{cmd}} - \omega_t}{\tau_\omega}\Delta t, \\
x_{t+1} &= x_t + v_{t+1}\cos(\theta_t)\,\Delta t, \\
y_{t+1} &= y_t + v_{t+1}\sin(\theta_t)\,\Delta t, \\
\theta_{t+1} &= \theta_t + \omega_{t+1}\,\Delta t.
\end{aligned}$$

Where τ_v =0.2 τ_ω=0.1. This setup balances model fidelity and computational simplicity

2.2 DWA Trajectory Prediction

At each candidate command (v, ω)

- (v, ω), the system predicts a short trajectory using the same discrete model, applying repeated updates until the predict time.
- This unifies local planning and execution kinematics, enabling cost computation (goal distance, heading, obstacle proximity) directly on predicted states. This mirrors standard DWA formulations.

2.3 Challenges

Key challenges include:

- Nonlinear kinematics due to trigonometric terms in Equations (2)–(6).
- Balancing goal-reaching with obstacle avoidance under real-time constraints.

- Ensuring computational efficiency for real-time implementation.

3 Path and Trajectory Planning

3.1 Path Planning – A* Global Planner

Grid Construction & Search: Create a 2D occupancy grid from circular obstacles (inflated by robot radius) and perform A* search on a uniform grid (0.25 m, including diagonal moves) using the Euclidean heuristic.

Path Simplification: Apply a modified Ramer–Douglas–Peucker algorithm to remove redundant waypoints, improving trajectory smoothness and execution efficiency.

Role: Provides a globally feasible, collision-free route from start to goal but assumes a fully static obstacle map.

3.2 Local Obstacle Avoidance – Dynamic Window Approach (DWA)

Velocity Sampling: Define a dynamic window of admissible (v, ω)

- (v, ω) based on current velocities and acceleration limits, simulating reachable velocities over a short time horizon.

3.3 Trajectory Prediction

For each candidate pair, predict future states using the same kinematic model employed by MPC.

3.4 Cost Evaluation

Score trajectories using a weighted objective:

$$\text{score} = w_g J_{\text{goal}} + w_s v + w_h J_{\text{heading}} - w_o J_{\text{obs}} - w_p |\Delta\theta_{\text{path}}|$$

3.5 Execution

Choose the highest-scoring, collision-free path and update at a fixed frequency. If no solution is found, trigger reverse + turn recovery behavior.

3.6 Trajectory Refinement – Model Predictive Control (MPC) Horizon & Dynamics

Solve over a two-step horizon (N=2), using the discrete kinematic model:

$$v_{k+1} = v_k + \frac{v_{\text{cmd},k} - v_k}{\tau_v}\Delta t, \qquad \tau_v = 0.2\,\text{s}$$
$$\omega_{k+1} = \omega_k + \frac{\omega_{\text{cmd},k} - \omega_k}{\tau_\omega}\Delta t, \quad \tau_\omega = 0.1\,\text{s}$$

$$x_{k+1} = x_k + v_{k+1}\cos\theta_k\Delta t, \quad y_{k+1} = y_k + v_{k+1}\sin\theta_k\Delta t, \quad \theta_{k+1} = \theta_k + \omega_{k+1}\Delta t$$

Cost Function:

$$\min_{v_k,\omega_k} \sum_{k=0}^{1} \Big[w_{\text{pos}}\|p_k - p_k^{\text{ref}}\| + w_{\text{head}}|\theta_k - \theta_k^{\text{ref}}| + w_{\text{obs}}J_{\text{obs},k}$$
$$+ w_{dv}(v_k - v_{k-1})^2 + w_{dw}(\omega_k - \omega_{k-1})^2 + w_v(v_k - v_{\text{DWA}})^2 + w_w(\omega_k - \omega_{\text{DWA}})^2 \Big]$$

3.7 Optimization

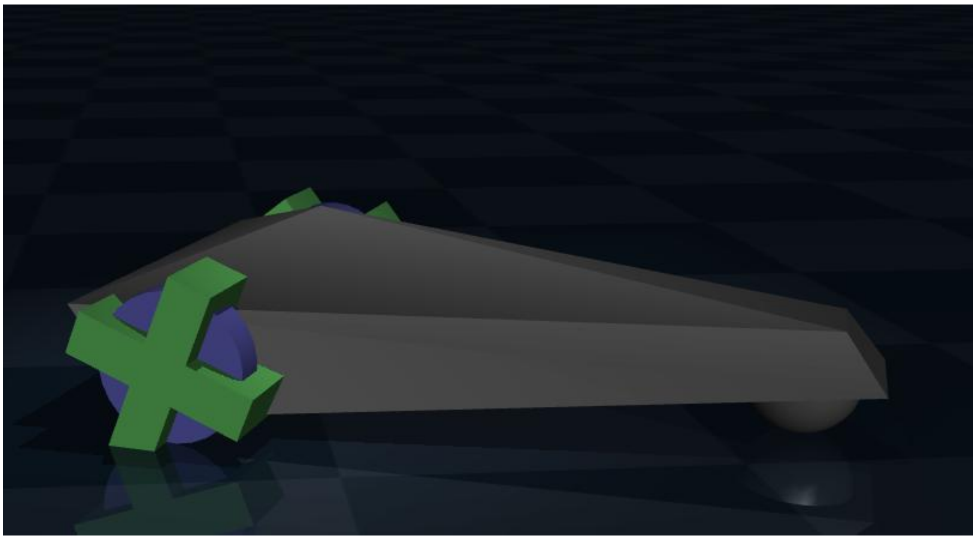
Implemented using SciPy's SLSQP solver. Despite nonlinearity, the small horizon makes real-time execution tractable.

3.8 Motion Execution – PID Control

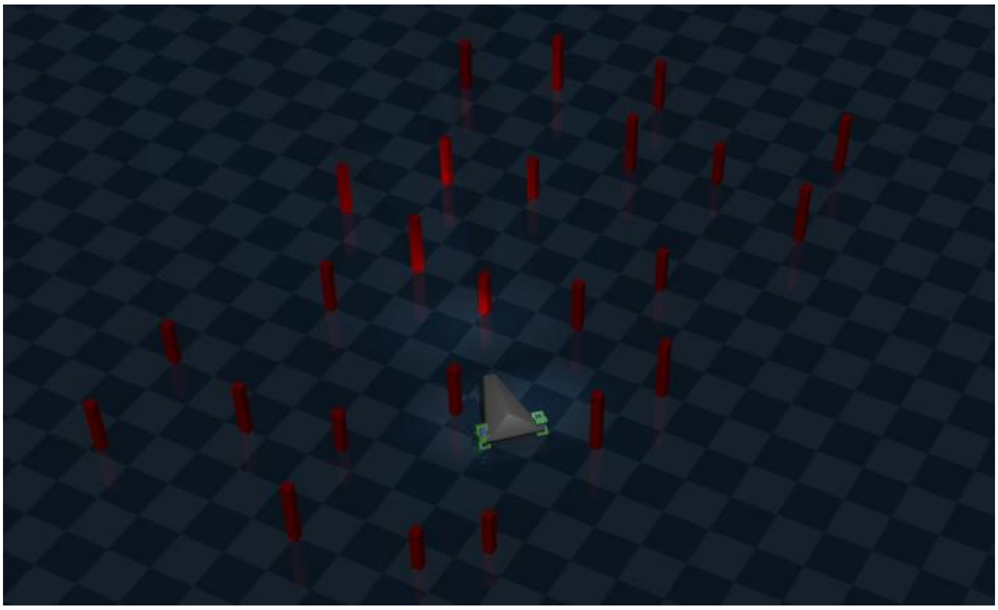
- **Low-Level Control:** Use classical PID loops to convert high-level commands (v, ω) into motor control signals.
- **Filtering:** Apply exponential smoothing (filter alpha) to reduce noise and improve actuator response.
- **Constraints**: Clip control commands within $[-1,1]$ to match actuator limits and maintain real-time stability.

4 Simulation and Results

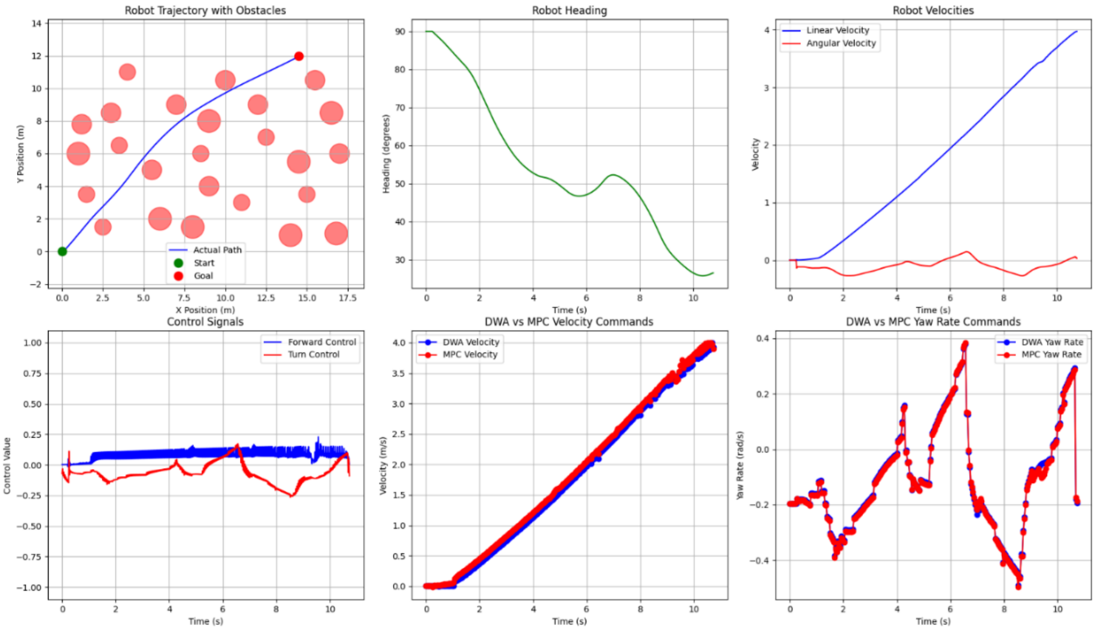
4.1 Differential Drive UGV



4.2 Environment Setup



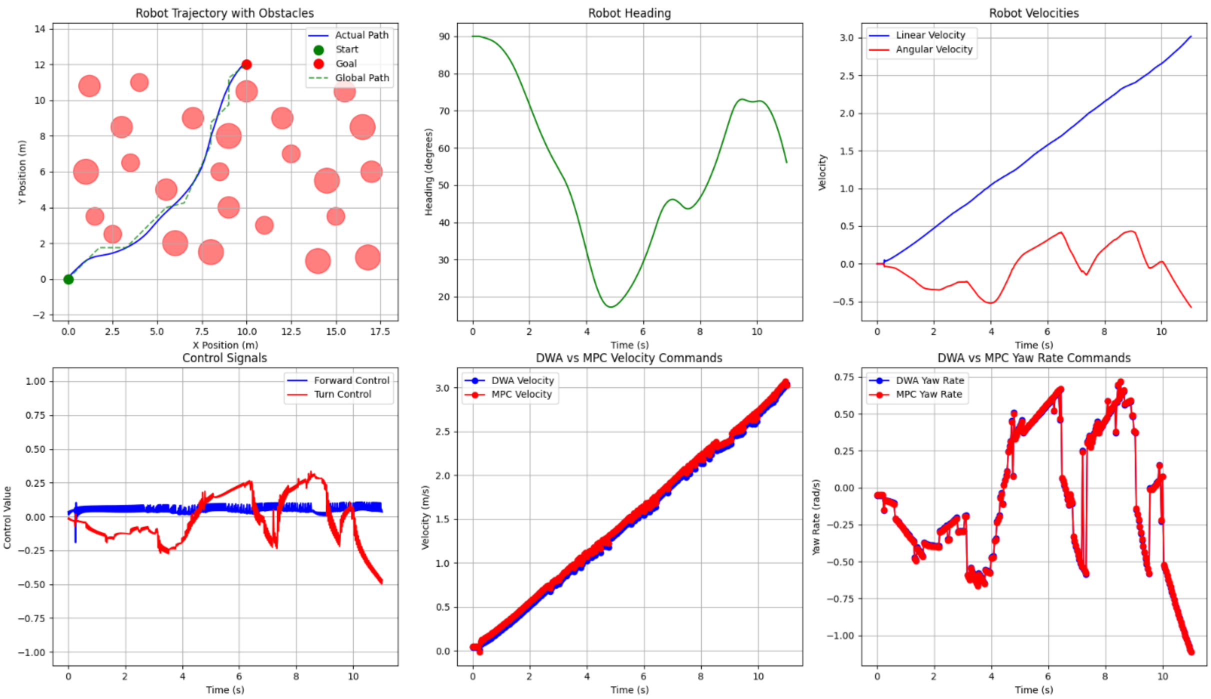
4.3 Result and Plotting



This is our first plot we obtained shows a mobile robot's successful obstacle-avoiding path (blue curve) from start (green dot) to goal (red star). The control graphs below reveal:

- **Linear Velocity:** Smoothly varies (0 - 0.6 m/s), indicating stable movement.
- **Angular Velocity:** Fluctuates (±1 rad/s), reflecting obstacle maneuvers.

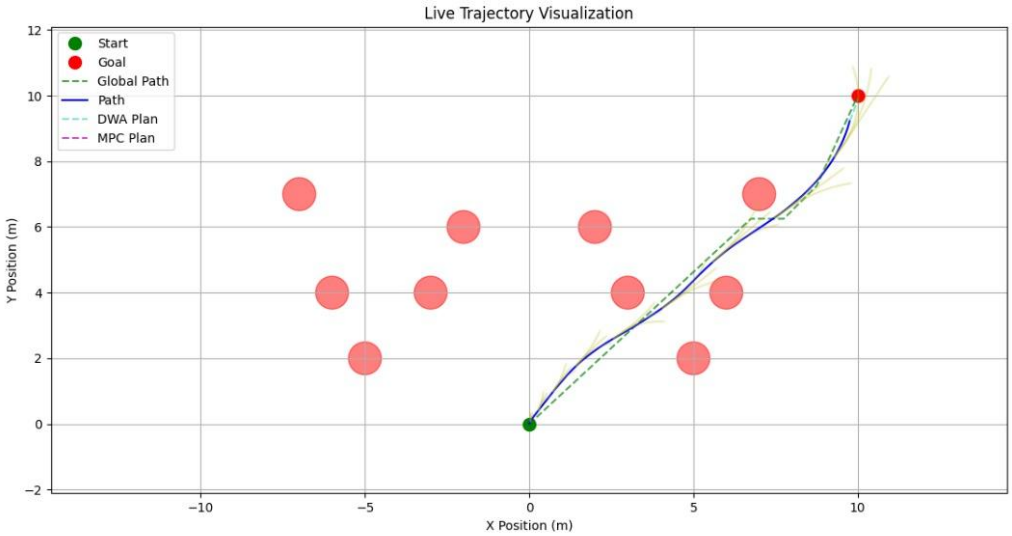
Comparing algorithms (DWA vs. MPC), MPC produces smoother control signals than the more reactive, fluctuating DWA. The robot achieved its goal effectively.



This is the better from the previous one it shows a smoother, more direct robot path (blue curve) from start (green circle) to goal (red star) Velocity graphs indicate:

- **Linear velocity:** Stable (0-0.5 m/s)

- **Angular Velocity:** Calmer (±1 rad/s) than previous result Likely using predictive planning MPC for optimized obstacle avoidance and efficient navigation.



In this plot multiple paths overlay in complex obstacles:

- Global: Coarse route
- DWA: Reactive deviations
- MPC: Smoother, closer to global
- Actually: Tracks MPC Precisely Shows successful integration of planning/control for optimized navigation.

4.4 Comparative Analysis

Architecture	Global Planning	Local Reactivity	Trajectory Smoothness	Computation Cost
A*	✓	✗	✗	Low
DWA	✗	✓	✗	Low–Medium
DWA + MPC	✗	✓	✓	Medium
A* + DWA + M	✓	✓	✓	Medium–High

5 Research Contribution

5.1 Relevance

• Hybrid Planner–Controller Architecture: The code employs a three-tiered navigation stack (A* → DWA → MPC → PID), ensuring global feasibility, reactive safety, and trajectory optimization using only on-board computation.

- Improved Robustness: Integrating MPC refines the reactive DWA commands, reducing oscillations and improving smoothness—demonstrated in the simulation's continuous velocity profiles and clean trajectory paths.

5.2 Validation

Simulation results validate the approach:

- Trajectory Tracking: The robot consistently reaches the goal with smooth and collision-free paths.

• Velocity Command Alignment: MPC outputs closely follow DWA suggestions, but with significantly less variation, as seen in overlay plots of command profiles.

5.3 Limitations & Future Work

• Static Environment Only: The current pipeline assumes fixed obstacles and requires manual replanning if the map changes.

• Short MPC Horizon: A short horizon limits long-term foresight.
• Odometry Not Addressed: Wheel slip and measurement drift are unmodeled.
• Future Enhancements:
  - o Integrate sensor-based dynamic replanning (e.g., new obstacle detection and map updates).
  - o Extend MPC horizon, possibly with faster solvers (QP-based).
  - o Add Control Barrier Functions or robust constraint management for formal safety.

• Implement state estimation (e.g., EKF) to manage odometric errors.

6 Conclusion

The experiments validate that combining a global planner (A*) with local planning (DWA) and motion optimization (MPC), executed via PID control, results in superior navigation performance for a differential drive robot. The robot achieves accurate goal-reaching with smooth, collision-free paths. Among the local planners, MPC consistently produces smoother trajectories and control signals compared to DWA, particularly when guided by a global path. The best performance—both visually and analytically—is observed when all modules are integrated, as shown in Result14. This confirms the effectiveness of the hierarchical planning and control architecture for real-time robotic navigation in complex environments.

References

[1] Chen, X., et al., A Hybrid DWA-MPC Framework for Coordinated Path Planning and Collision Avoidance in Articulated Steering Vehicles. Machines, 2024. 12(12): p. 939.

[2] Rodríguez-Molina, A., Herroz-Herrera, A., Aldape-Pérez, M., Flores-Caballero, G., & Antón-Vargas, J. A. (2022). Dynamic Path Planning for the Differential Drive Mobile Robot Based on Online Metaheuristic Optimization. Mathematics, 10(21), 3990. https://doi.org/10.3390/math10213990.

[3] Chen, J., & Cheng, J. (2024). Study of Robot Path Planning with Improved A* And DWA Algorithm Fusion. Academic Journal of Science and Technology, 9(2), 77-82.

[4] K. Li, D. Zhang, X. Li and Y. Su, "A Study of Mobile Robot Path Planning by Fusing Improved A* And DWA Algorithms," 2023 China Automation Congress (CAC), Chongqing, China, 2023, pp. 2242-2247, doi: 10.1109/CAC59555.2023.10450791.

[5] Šelek, A., M. Seder, and I. Petrović, Smooth Autonomous Patrolling for a Differential-Drive Mobile Robot in Dynamic Environments. Sensors, 2023. 23(17): p. 7421.