Object path modeling with external disturbance compensation and adaptive replanning*

Yurii Zhyshchynskyi[1,†], Serhiy Horiashchenko[2,†] and Yehor Solomianyi[2,†]

[1] *Kherson State Maritime Academy, 20, Ushakov ave., 73000, Kherson, Ukraine*
[2] *Khmelnytskyi National University, Institutska str., 11, Khmelnytskyi, 29016, Ukraine*

Abstract

This study introduces a wind-resilient path planning methodology based on an enhanced "Hybrid Way" algorithm with integrated real-time replanning capabilities. Unlike conventional implementations that neglect environmental disturbances, the proposed framework explicitly incorporates a wind force model into the state transition and cost evaluation functions. The algorithm dynamically adjusts steering angles, motion primitives, and cost weights to mitigate wind-induced trajectory deviations. Furthermore, a continuous replanning mechanism is employed to ensure path optimality and feasibility under rapidly changing wind conditions. The proposed approach is validated through a series of simulation experiments in Python, where it consistently outperforms the baseline "Hybrid Way" in terms of deviation reduction, path smoothness, and travel cost efficiency. The results confirm the method's potential for autonomous ground, marine and aerial vehicles operating in outdoor environments with significant dynamic disturbances.

Keywords

path planning; wind disturbance; real-time replanning; autonomous navigation; environmental uncertainty; dynamic obstacles; trajectory optimization; simulation; algorithms, Python.

1. Introduction

Automated systems for computing the trajectory of objects depend on intricate mathematical models and algorithms that consider numerous factors influencing the object's motion. These encompass both external factors, such as wind, gravity, weather, and the object's inherent qualities, including its mass, size, aerodynamic traits, etc. [1] Developing these systems necessitates extensive knowledge in mathematics, physics, computer science, and engineering, alongside experience in resolving practical navigation and control challenges. A key objective of automated systems is to guarantee precise trajectory calculations [2]. This is especially crucial when the object is in challenging environments or undertaking vital tasks. For instance, in astronautics, proper trajectory calculation is essential for successful orbit entry, maneuvers, and return to Earth. In the transportation sector, precise trajectory planning helps avert accidents, ensure passenger security, and optimize fuel use. Automated systems for calculating object trajectories have widespread applications across various industries. In transport, such systems are employed to manage autopilots, unmanned aerial vehicles, rail systems, and marine vessels. In aviation, they offer accurate flight route planning, automatic aircraft control, and air traffic management. In astronautics, they're used to compute orbital maneuvers, control spacecraft, and plan missions. In robotics, such systems permit robots to effectively perform intricate tasks in a dynamic environment.

*ITTAP'2025: 5th International Workshop on Information Technologies: Theoretical and Applied Problems, October 22-24, 2025, Ternopil, Ukraine, Opole, Poland.

[1*] Corresponding author.
[†] These authors contributed equally.
jekas1367@gmail.com (Y. Zhyshchynskyi); tnt7@ukr.net (S. Horiashchenko); solomyanyyy@gmail.com (Ye. Solomianyi
0009-0007-4020-4576 (Y. Zhyshchynskyi); 0000-0001-6623-2523 (S. Horiashchenko); 0009-0000-7553-2763 (Ye. Solomianyi)

Automating motion calculation procedures permits lessening human involvement in commonplace and hazardous tasks, boosting the precision and swiftness of task execution, and diminishing the chance of mistakes. This unveils fresh opportunities for technological advancement and the betterment of people's lives. Nonetheless, the evolution of such systems demands considerable efforts and assets, in addition to a complete strategy for tackling automation challenges [2, 3]. The significance of the research subject stems from several key elements. Initially, the automation of vehicle management procedures, for example automobiles, unmanned aerial vehicles, and maritime vessels, demands precise and speedy trajectory computations to ensure the safety and effectiveness of movement [4]. Secondly, within robotics, the exactness of trajectory calculations establishes the ability of robots to execute complex manipulations in real time, which is crucial for industrial and service robots in daily routines. Third, in military affairs, accurate prediction of the trajectory of objects allows to ensure the effectiveness of combat operations and minimize risks.

2. Presenting the main material

Modern automated trajectory calculation systems use a wide range of mathematical models, including classical methods of mechanics, as well as newer approaches based on artificial intelligence methods. Classical methods include Newton's equations, which describe the motion of objects under the influence of forces. Modern approaches include machine learning methods, in particular neural networks, which are able to learn from large amounts of data and make predictions with high accuracy [2].

Dynamic models of moving entities are crucial for comprehending and forecasting their conduct across diverse surroundings. These models enable us to depict the motion of an entity while considering various physical and mechanical parameters, such as mass, force, acceleration, environmental resistance, and so on. The advancement and application of dynamic models is very significant in many areas of science and technology, including transport, aviation, astronautics, robotics and other areas [3]. Dynamic models are mathematical or numerical depictions of object motion that consider both the internal characteristics of the objects and the external impacts influencing them. They aid in understanding how an object will move under the effect of specific forces and how its position, velocity, and acceleration will alter over time. Dynamic models rely on the principles of mechanics, specifically, Newton's three laws, which govern the interplay between forces and object motion. To precisely portray the movement of objects in real scenarios, it is essential to consider numerous elements that impact their motion. These could be both internal properties of the object, such as its mass, dimensions, shape, and external forces, such as gravity, aerodynamic drag, friction, magnetic fields, etc. Furthermore, it's crucial to consider the conditions of the surrounding space, which may evolve over time, such as temperature, humidity, pressure, etc. All these factors together determine the complexity and exactness of dynamic models [5].

In the transportation industry, dynamic models are used to analyze and optimize the movement of various vehicles.

Let us define the position: $x=(x, y)$, the velocity $v=\dot{x}$, the direction (course) θ and the angular velocity $\omega=\dot{\theta}^{\cdot}$. The mass is m.

Basic equations of motion:

$$m\dot{v}=F_{drive}+F_{drag}+F_{fric}+F_{wind}+F_{obs} \quad (1)$$

where

F_{drive} is the control force (from the engine/accelerator), directional — along the course θ;

$F_{drive} = u_t \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$, where $u_t$is a scalar (force control).

$F_{drag} = -c_d \| v_{rel} \| v_{rel}$ - the nonlinear aerodynamic drag,

where $v_{rel} = v - w(x,t)$ - relative velocity relative to air, c_d is the coefficient.

$F_{fric} = \mu(x,t) mg \frac{v}{\| v \| + \varepsilon}$, is the dry friction / sliding along the surface; $\mu(x,t)$ is the space-time friction coefficient, ε - is the regularizer at zero speed.

F_{wind} is added as the force from the wind, but if we take into account through v_{rel} in drag, the explicit addition may be immeasurable; for strong gusts, a stochastic component can be added.

$F_{obsm}$is the force of obstacle avoidance.

Angular dynamics (rotation / steering) for a mobile body with limited angular acceleration [6]:

$$I \dot{\omega} = \tau_{steer} - c_\omega \omega \dot{\theta} = \omega \qquad (2)$$

where I is the moment of inertia, $\tau_{steer}$is the steering torque (control u_θ), c_ω is the damping.

Wind is the vector field $w(x,t)$.

We will use stochastic model: gusts as an Ornstein–Uhlenbeck (OU) process [7]:

$$dw = -\kappa(w - \overline{w}(x))dt + \sigma d W_t \qquad (3)$$

This gives time-correlated gusts with amplitude σ and relaxation κ.

Variable friction model [9]:

$$(x,t) = \mu 0 + \Delta\mu(x) + \eta(t) \qquad (4)$$

$\Delta\mu(x)$ is a surface map; $\eta(t)$ is a stochastic fluctuation (dust/moisture).

Obstacles and detours have three approaches [10, 11]:

1. Repulsion potentials: each static/moving obstacle sets a repulsive field $U_{rep}(r)U$, where r is the distance to the obstacle; force: $Fobs = -\nabla U_{rep}$. Simple and smooth, but prone to local minima.

2. Velocity Obstacles / Reciprocal Velocity Obstacles (RVO): for dynamic avoidance at the speed level (calculate the set of speeds that will lead to a collision).

3. Local controller - Dynamic Window Approach (DWA) or Model Predictive Control (MPC): at each step we optimize the control (u_t, u_\theta) for a short horizon taking into account the dynamics and obstacles. MPC gives better results, but is more computationally expensive.

Safety and speed constraints:

$$0 \le // v // \le v_{max}(x,t), / ut / \le U_{max}, / \tau_{steer} / \le T_{max} \tag{5}$$

where $v_{max}(x,t)$can decrease in the area of obstacles or poor friction.

Example of specific equations (reduced system)

Let us denote the state $s=[x, y, v_x, v_y, \theta, \omega]^T$

Then

$$\begin{cases} \dot{x}=v_x \\ \dot{y}=v_y \\ \dot{v}x=\frac{1}{m}\left(u_t cos\theta - c_d // v - w // (v_x - w_x) - \mu(x,t) mg \frac{vx}{// v // + \varepsilon} + F_{obs,x}\right), \\ \dot{v}y=\frac{1}{m}\left(u_t sin\theta - c_d // v - w // (v_x - w_x) - \mu(x,t) mg \frac{vx}{// v // + \varepsilon} + F_{obs,y}\right), \\ \dot{\theta}=\omega, \\ \dot{\omega}=\frac{1}{I}(T_{steer} - c_\omega \omega). \end{cases} \tag{6}$$

Calculation of the repulsive force:

$$U_{rep}(d)=\begin{cases} \frac{1}{2}k_{rep}\left(\frac{1}{d}-\frac{1}{d_0}\right)^2, d<d_0, F_{obs}=-\nabla U_{rep} \\ 0, d \ge d_0 \end{cases} \tag{7}$$

In parallel, perform collision checking and, if the repulsive approach is stuck, run a local planner to rebuild the trajectory.

For evaluation, use Monte-Carlo simulations or Kalman filter / partial filter (EKF/UKF/Particle) to estimate the state during movement.

For deterministic simulations: RK4 with adaptive step dt; for rigid members (strong friction) — implicit Euler or semi-implicit scheme. Integration step: choose so that the Courant condition is fulfilled; for example dt = 0.01…0.1 s for car-like movement.

Event detection: when approaching an obstacle, perform a check and switch the controller.

Test extreme cases: zero speed, a sharp gust of wind, a sharp decrease in μ.

3. Development and testing of a program for path optimization

The developed simulator models the motion of a body along a complex 2D route while considering:

– Wind influence (direction and strength vary over time)

- Surface friction (affects acceleration)
- Obstacles (rectangular, requiring detours)
- Dynamic replanning of the path when conditions change

The simulation uses an enhanced “Hybrid way” path-planning algorithm, incorporating both kinematic constraints and environmental disturbances. The program records both the planned trajectory (ideal path from “Hybrid way”) and the actual trajectory (affected by wind, obstacles, and dynamics) for comparison.

The “Hybrid way” algorithm is used to find a path from the start point to the goal.

“Hybrid way” considers the orientation of the vehicle/body and continuous state space, allowing for realistic turn constraints and smooth paths. The cost function includes:

- Euclidean distance to the goal (heuristic);
- Turning penalties;
- Obstacle avoidance cost;
- Path smoothness factor.

The simulation proceeds in discrete time steps. At each step the current wind vector is generated or updated (variable wind). The body’s motion is updated using Newtonian dynamics. Collision detection is performed with obstacles.

If a collision or significant deviation occurs, “Hybrid way” is re-run from the current position to the goal. Movement continues toward the next waypoint [12].

The program records a CSV log for every run:

- Real motion (affected by wind, collisions, and replanning);
- Planned motion (original “Hybrid Way” path without disturbances).

This allows post-simulation analysis of path deviations, time delays, and energy cost.

We used Libraries such as “Pygame” for real-time visualization of the simulation. Handles rendering of the environment, obstacles, wind vector display, and body movement. Provides a game loop structure for smooth animations.

In addition, we used “math” (Python Standard Library) for trigonometric calculations (sin, cos, atan) and coordinate transformations.

For generates variability in wind direction and strength “random” was used. Randomly positions obstacles within constraints.

Saves simulation data into .csv files for later analysis. Each log includes time steps, positions, orientations, velocities, wind parameters, and replanning flags.

The program code is written in Python. Parts of the program code, such as the start and parameter input, are shown in Figure 1.

```python
# hybrid_astar_with_logs.py
import os
import tkinter as tk
from tkinter import messagebox
import pygame
import sys
import random
import math
import csv
import time
from heapq import heappush, heappop
from datetime import datetime
```

```python
def run_simulation(params):
    # unpack params
    mass = params["mass"]
    vmax = params["vmax"]
    map_w = params["map_w"]
    map_h = params["map_h"]
    n_obs = params["n_obs"]
    obs_min = params["obs_min"]
    obs_max = params["obs_max"]
    wind_max = params["wind_max"]
    wind_interval = params["wind_interval"]
    cell_size = params["cell_size"]
    n_headings = params["n_headings"]
```

Figure 1: Code Parts: Starting the Code and Entering Parameters

When you start the program, a dialog box opens where we can set the parameters for modeling the object's motion. For example, a 500 kg boat is chosen that has to swim 800 meters. There may be obstacles and wind in the boat's path.

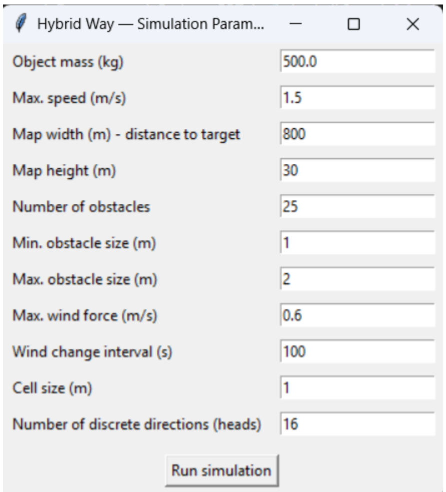


Figure 2: The dialog box

After starting the simulation, we will see a field where the object moves and random obstacles that occur on its path. In addition, the object is affected by a changing wind force. The goal is to find the optimal path to the final point. The beginning of the movement of the object is shown in Fig. 3. Black line on object - direction of movement taking into account the erase action.

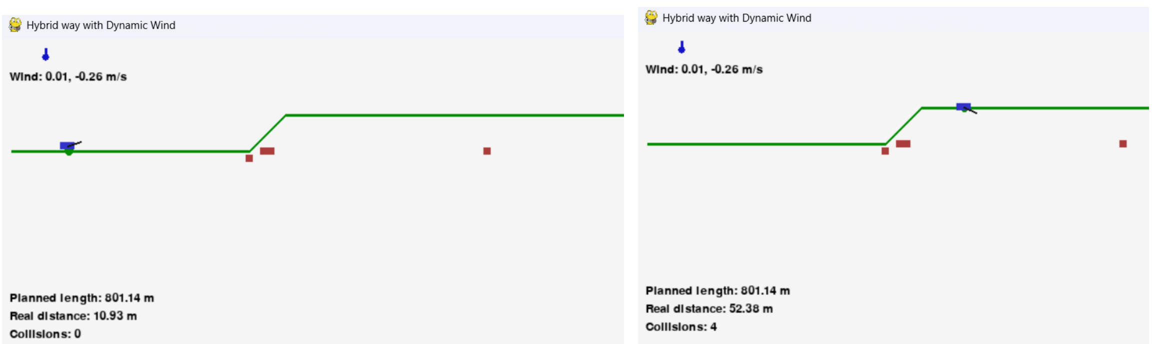
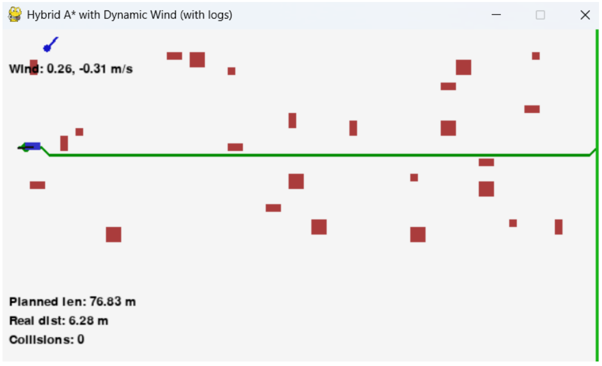


Figure 3: The Motion simulation window

The end point of the path for the object is shown in Fig. 4. It shows random obstacles and the green line is the probable optimal route of the object. The simulation result is shown in a separate window in figure 5.



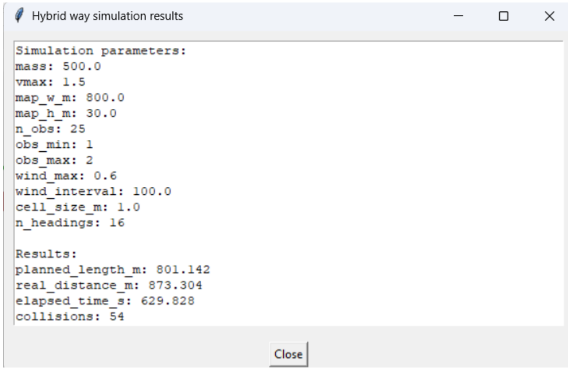


Figure 4: The end point of the path **Figure 5:** Results of modelling

As simulation experiments have shown, critical situations are possible when an object needs to avoid an obstacle. The critical situations that arose are shown in Fig. 6 when the wind moved the object to the transition.

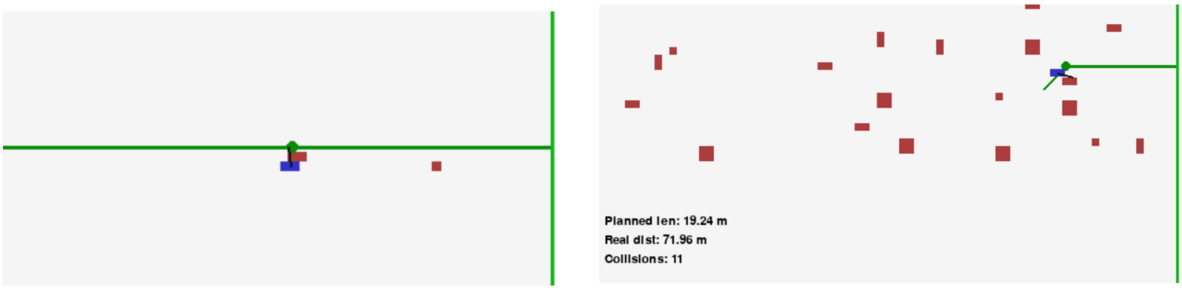
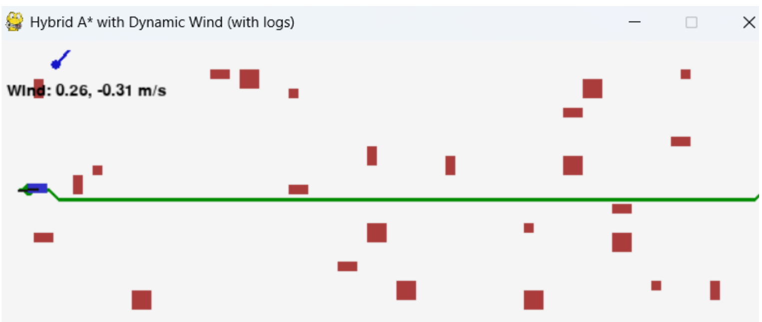


Figure 6: Direct collision with an obstacle due to wind

After optimization and adding a safe distance value, we were able to get the object to move without collisions. An example is shown in Fig. 7 and the simulation result is shown in Fig.8




```
mass: 500.0
vmax: 1.5
map_w_m: 800.0
map_h_m: 30.0
n_obs: 25
obs_min: 1
obs_max: 2
wind_max: 0.6
wind_interval: 3.0
cell_size_m: 1.0
n_headings: 16

Результати:
planned_length_m: 3.0
real_distance_m: 1454.29
elapsed_time_s: 982.965
collisions: 0
avg_wind_m_s: 0.303
max_wind_m_s: 0.6
```

Figure 7: Optimal object movement value **Figure 8:** Results of modelling

Advantages of this approach are given realism then Incorporates continuous dynamics, disturbances, and physical constraints. In addition, we get flexibility then supports dynamic obstacle avoidance and environmental changes and visualization is useful for real-time graphical display helps debug and analyze behavior.

This simulation framework can be adapted for autonomous ground vehicle navigation, UAV (drone) path planning under wind disturbances and Robotics motion planning in cluttered environments. Research in optimal control and adaptive navigation strategies.

4. Conclusions

In general, dynamic models of moving objects are a powerful tool for engineers and scientists working on the development of new technologies and systems. They allow for a deeper understanding of the physical principles underlying motion and the use of this knowledge to solve practical problems. With the development of computing and modern modeling methods, the capabilities of dynamic models are constantly expanding, opening up new prospects for research and innovation. The development and improvement of dynamic models is an important area of scientific research that will contribute to further progress in the creation of new technologies and systems that improve the quality of life and safety of people.

Development and evaluation of a "Hybrid Way" path planning framework integrated with real-time replanning capabilities under wind disturbance conditions was created. The proposed approach effectively combines the deterministic search characteristics of "Hybrid Way" with adaptive trajectory updates, enabling the system to maintain feasible and efficient navigation in dynamic and uncertain environments.

Despite the fact that the object traveled 1.8 times longer than planned, we did not receive any collisions. Simulation results in demonstrate that the method achieves robust path tracking while minimizing deviations caused by lateral wind forces. The incorporation of environmental feedback allows the planner to proactively adjust trajectories, ensuring collision avoidance and maintaining operational safety. Compared to static planning, the hybrid approach exhibits superior adaptability, particularly in scenarios with fluctuating environmental conditions and dynamic obstacles.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] Y. Haartsen, R. Aalmoes and Y. S. Cheung, "Simulation of unmanned aerial vehicles in the determination of accident locations," *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, Arlington, VA, USA, 2016, pp. 993-1002, doi: 10.1109/ICUAS.2016.7502548.

[2] O. Matsebe, C.M. Kumile, N.S. Tlale, A Review of Virtual Simulators for Autonomous Underwater Vehicles (AUVs), IFAC Proceedings Volumes, Volume 41, Issue 1, 2008, Pages 31-37, ISSN 1474-6670, ISBN 9783902661357, https://doi.org/10.3182/20080408-3-IE-4914.00007.

[3] Dolgov, D., Thrun, S., Montemerlo, M., & Diebel, J. (2008). Practical search techniques in path planning for autonomous driving. *Annals of Mathematics and Artificial Intelligence, 55*(3–4), 293–316. https://ai.stanford.edu/~ddolgov/papers/dolgov_gpp_stair08.pdf

[4] Pivtoraiko, M., & Kelly, A. (2005). Efficient constrained path planning via search in state lattices. In *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)* (pp. 1–8). https://www.cs.cmu.edu/~alonzo/pubs/papers/isairas05Planning.pdf

[5] H. C. Pangborn, M. A. Williams, J. P. Koeln and A. G. Alleyne, "Graph-based hierarchical control of thermal-fluid power flow systems," *2017 American Control Conference (ACC)*, Seattle, WA, USA, 2017, pp. 2099-2105, doi: 10.23919/ACC.2017.7963262.

[6] Kelly A, Nagy B. Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control. *The International Journal of Robotics Research*. 2003;22(7-8):583-601. doi:10.1177/02783649030227008

[7] LaValle, S. M. (2006). *Planning algorithms.* Cambridge University Press. https://doi.org/10.1017/CBO9780511546877

[8] Kelly, A., & Nagy, B. (2003). Reactive nonholonomic trajectory generation via parametric optimal control. *The International Journal of Robotics Research, 22*(7–8), 583–601. https://www.cs.cmu.edu/~alonzo/pubs/papers/ijrr02TrajGen.pdf

[9] Cora ÖN, Akkök M, Darendeliler H. Modelling of variable friction in cold forging. Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology. 2008;222(7):899-908. Fossen, T. I. (2011). *Handbook of marine craft hydrodynamics and motion control.* John Wiley & Sons. https://doi.org/10.1002/9781119994138

[10] Automatic diagnostic device with measurement of distances to damages by the combined pulse-phase method Janusz Musiał, Kostyantin Horiashchenko, Serhiy Horiashchenko and Mikołaj Szyca, MATEC Web of Conferences 351, 01008 (2021) https://doi.org/10.1051/matecconf/202135101010

[11] S. Horiashchenko, O. Paraska, K. Horiashchenko, V. Onofriichuk, O. Synyuk and V. Pavlenko, "Mechatronic System for Management and Control of a Device for Applying a Polymer Coating," 2023 IEEE 5th International Conference on Modern Electrical and Energy System (MEES), Kremenchuk, Ukraine, 2023, pp. 1-5, doi: 10.1109/MEES61502.2023.10402550

[12] Arslan, M. (2016). *Obstacle detection and pathfinding for mobile robots* (Master's thesis, Near East University, Graduate School of Applied Sciences). Retrieved from https://vixra.org/pdf/1705.0027v1.pdf