Motion planning with obstacle avoidance for a trident snake robot

Szymon Furmańczyk

February 1, 2022

Intermediate project

under the direction of
Witold Paluszyński, PhD

Department of Cybernetics and Robotics,
Faculty of Electronics, Photonics and Microsystems,
Wrocław University of Science and Technology

Abstract

The goal of this project was to develop and test an algorithm for pathfinding with obstacle avoidance for a trident snake robot. In presented approach the problem was decomposed to two levels. On the high-level, path to the goal in the environment was found using the potential field method. On the low-level, motion of the robot was generated with the use of Jacobian motion planning algorithm. Results of implemented solution were verified and illustrated with use of computer simulations in MATLAB and CoppeliaSim.

1 Introduction

Trident snake robot is a snake-like robot developed for research purposes concerning motion planning and control algorithms in constrained non-holonomic systems. Due to complexity of singular configurations and possible collisions between parts of the robot, this task is quite challenging and computationally heavy.

1.1 Robot structure

In this project, trident snake robot equipped with active joints (undulatory locomotion) and arms consisting of single link (with passive wheel) was used. This configuration is presented on image below.

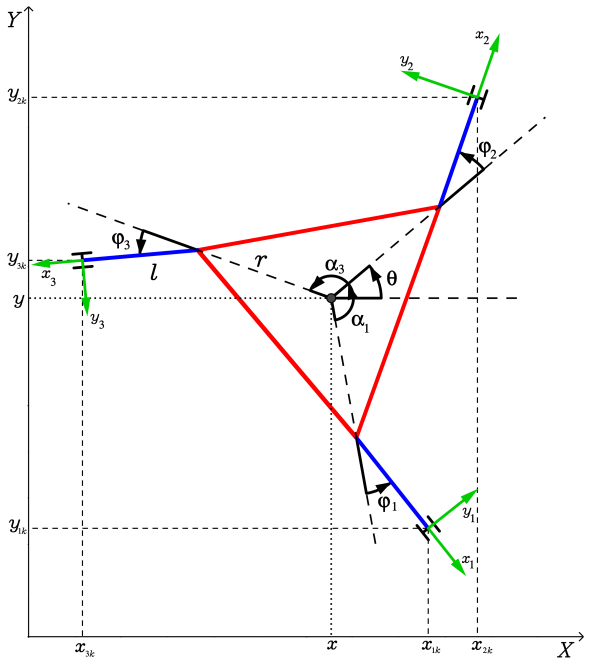


Figure 1: Geometric structure of a trident snake robot considered in this project

We can write generalized coordinates of this robot: $q = (x, y, \theta, \phi_1, \phi_2, \phi_3) \in R^6$, where:

- x, y - coordinates on a movement plane,
- θ - heading of robot,
- ϕ_1, ϕ_2, ϕ_3 - arms angular displacement.

To extend possibilities of analysis, model of the robot was developed in 3D modeling software, which was then used in simulations in physics engine.
Elementary components of the robot were modeled from drawings which were provided by distributors of used parts.
Additional models of hardware components (screws,nuts) were obtained from "McMaster-Carr" website [1].
Below, renders of models of robot body, arm and wheel are presented.

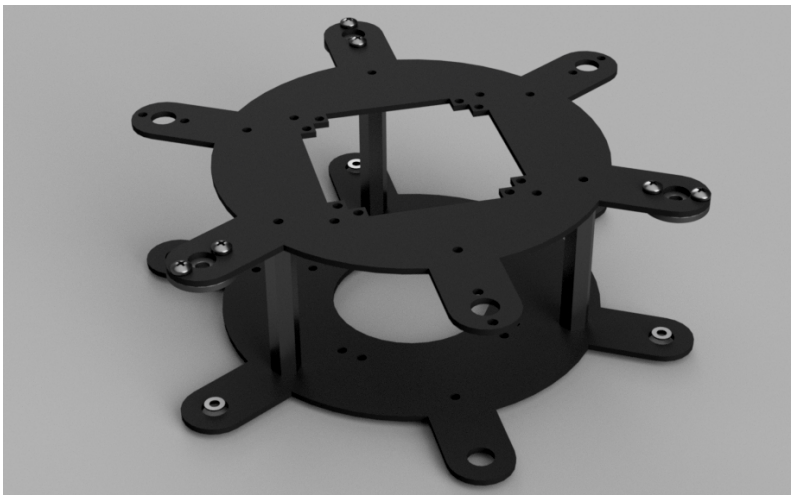


Figure 2: Body assembly

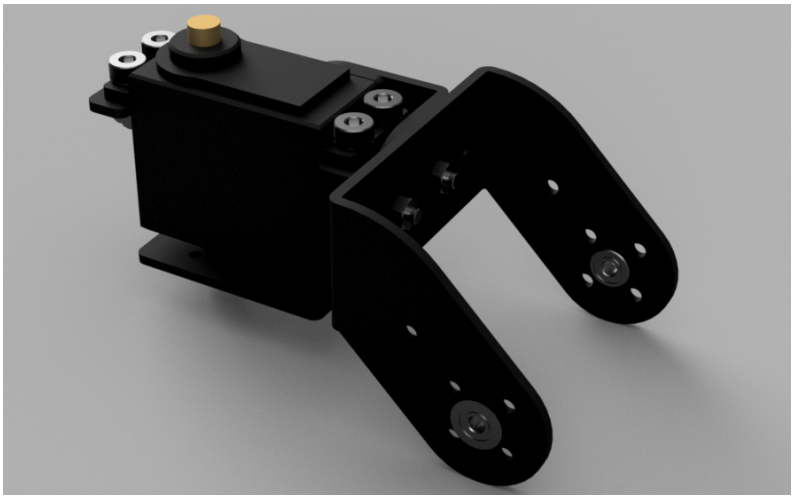


Figure 3: Arm assembly

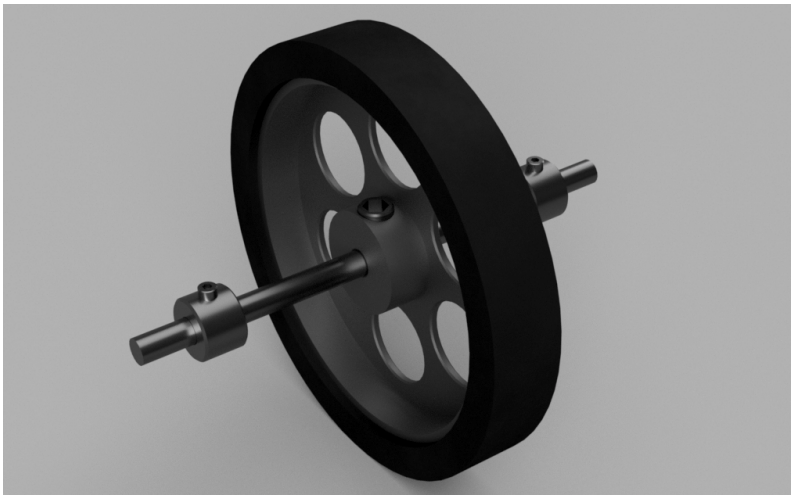


Figure 4: Wheel assembly

2 Description of motion planning problem and proposed solution

2.1 Motion planning task

During this project it was assumed that there was no lateral slip of the wheels.
The formal principles of motion of the robot were presented in depth in [2] and [3]. In contrary to that approach, where the movement is decomposed to translation and rotation, due to use of Endogenous Configuration Space Approach (ECSA) and singularity robust Jacobian inverse kinematics algorithm [4], it was possible to change position and orientation of the robot at once, as it has been done in [5].
As a contribution to the last source, the *smax* (smooth maximum) function has been replaced with:

$$smax(\iota,\mu) = \frac{\iota + \sqrt{\iota^2 + \mu}}{2}, \tag{1}$$

as it was giving more robust performance.
Example of movement is presented on the graph below.

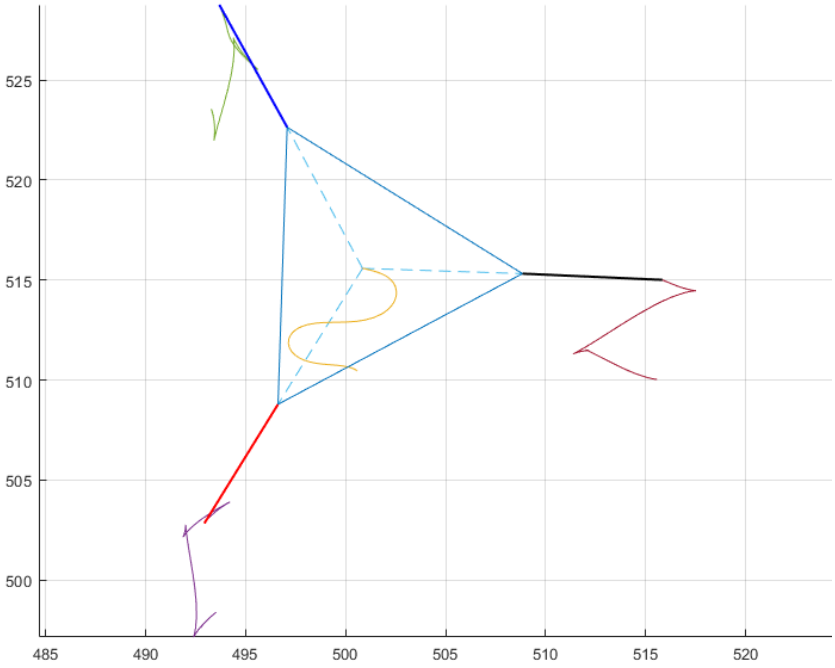


Figure 5: Movement of the robot on a short path

We can observe snake-like motion of center point of the robot and the paths which the wheels took during the movement.
In case of longer paths it is possible to decompose them to multiple shorter segments, for which computational complexity was significantly lower and allowed for iterative execution of the algorithm.
Example of movement on such path is presented on the graph below.

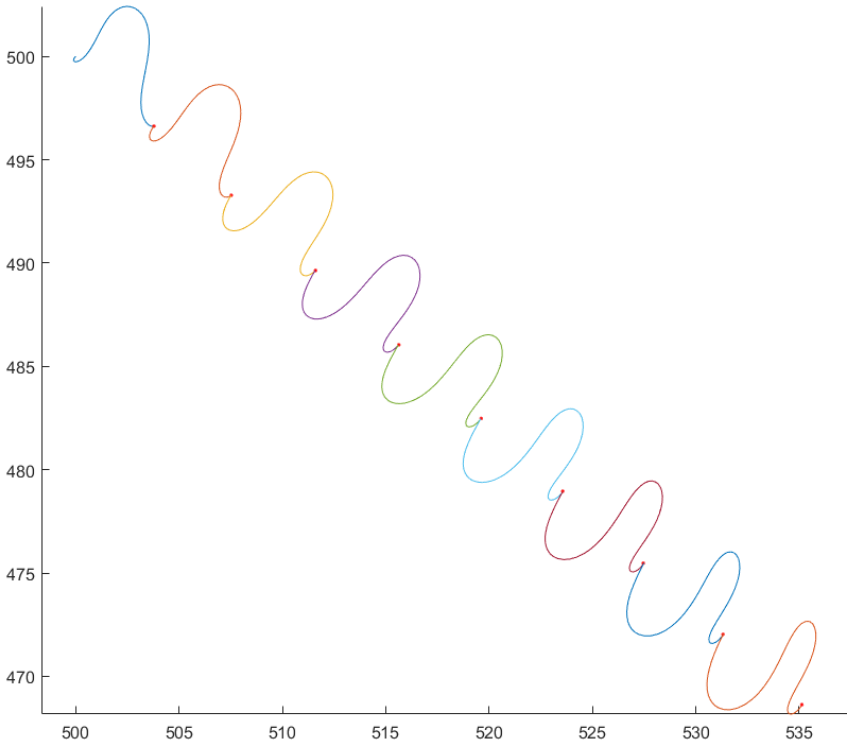


Figure 6: Movement of the robot on a long path

2.2 Obstacle avoidance

The obstacle avoidance was performed with the use of potential field method (similar to that presented in [6]).
At the start of the motion planning, potential field was generated by adding to each other these components:

- for each of the obstacles repulsive force was generated by means of function that has value close to zero far from the obstacle and some big value close to it, with additional certain transition threshold,
- for entire field attractive force is generated by means of convex function which has minimum in the position of the goal,
- repulsive force is generated by means of function which has big values close to the boundary of the field and zero otherwise.

All of the functions mentioned above had to be smooth.
With correct parameters and functions it was possible to obtain usable potential field, for example:

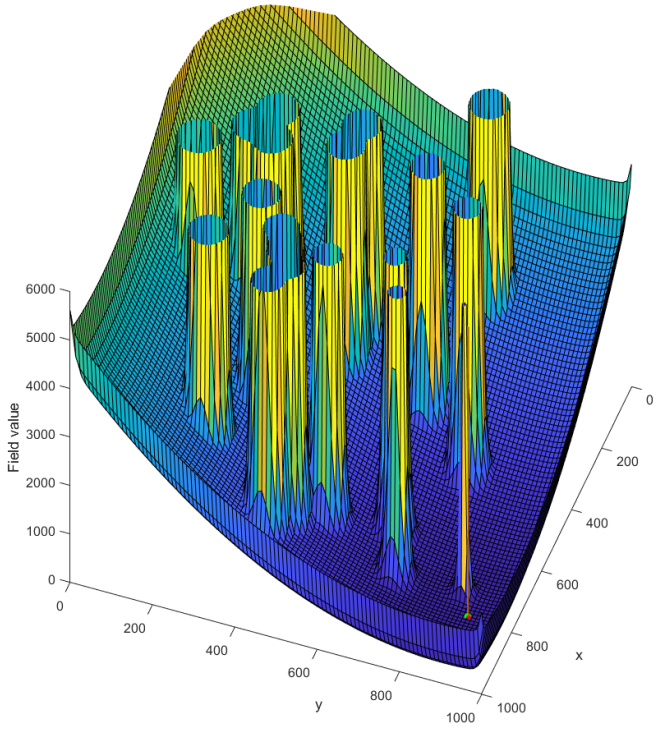
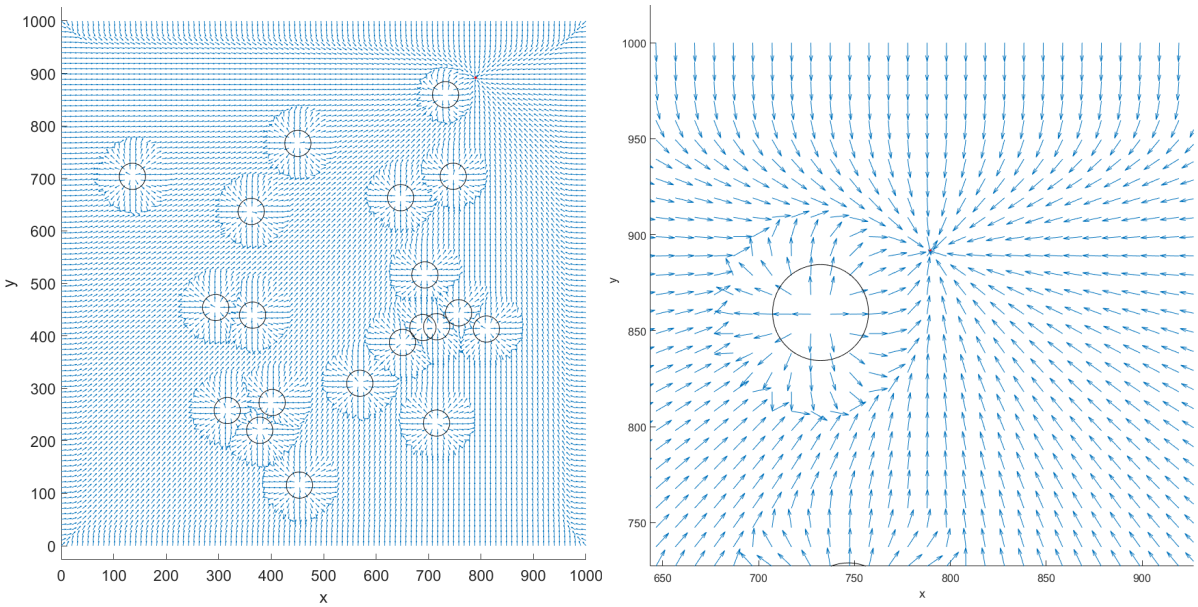


Figure 7: Example potential field

Direction of steepest descent for this field is shown on the graphs below.

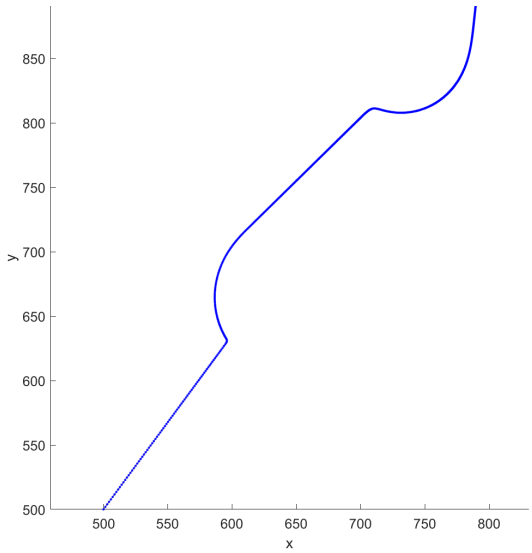


(a) Entire field

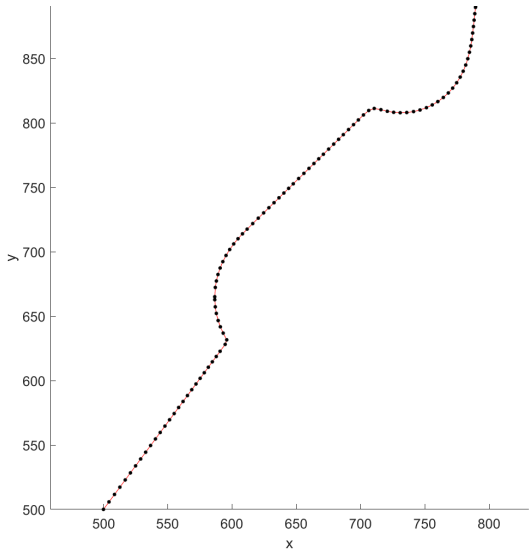
(b) Closer view

Figure 8: Direction of descent in sample points of the field

We can see repulsive behavior of obstacle (black circle) and attractive behavior of the goal (red point).
To obtain path to the goal from any position inside the field, gradient descent was used.
Example of computed path is shown below along with line segments that were obtained form it. These segments are of certain length that was dependent on the size of robot and desired accuracy of reconstruction of path generated by steepest descent algorithm.



(a) Path obtained from gradient descent



(b) Line segments obtained for the gradient descent path

Figure 9: Examples of the high-level motion planning

3 Overview of the implementation

3.1 Implementation in Matlab

In first step, potential field was generated for desired: number of obstacles, field size, size of obstacles, influence of obstacles and goal point position. It was assumed that initial position of the robot was in the center of the field.
In the next step, path from the position of the robot to the goal position was computed with gradient descent method and then segmented to allow for execution of low-level motion planning. Conditionally some precomputation took place (as to set up some necessary equations and speed up the algorithm).
Following that, connection of MATLAB and CoppeliaSim was established (following the guidelines of [7]).
Ultimately, iterative execution of low-level motion planning algorithm was performed and resulting Fourier series coefficients were sent to CoppeliaSim for each segment of the path obtained from high-level motion planning.
The computed coefficients were reused in successive iterations, which allowed for speeding up the computations when the segments were relatively similar (same direction and length).

3.2 Implementation in CoppeliaSim

To transfer the model from 3D modeling software to the CoppeliaSim, Unified Robot Description (URDF) file has been written. Inside this file, general structure and values of parameters of parts of the robot were specified (according to guidelines available on website [8]).
Resulting model is presented on the image below.

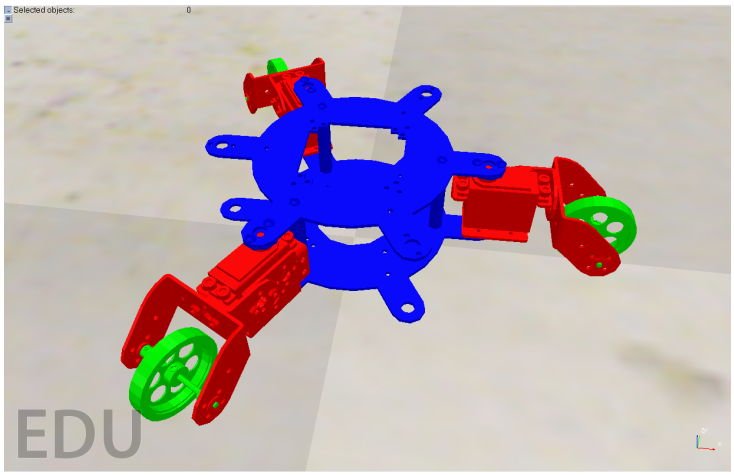


Figure 10: Robot model imported to CoppeliaSim with use of URDF file

To prepare the scene inside CoppeliaSim the data containing positions and size of obstacles was sent from MATLAB. This step was possible with the use of application programming interface delivered together with CoppeliaSim software.
Example of one of generated obstacles is presented on the image below.

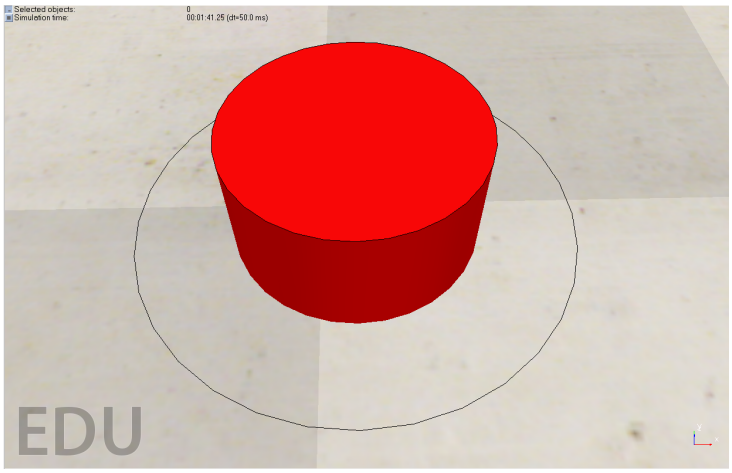
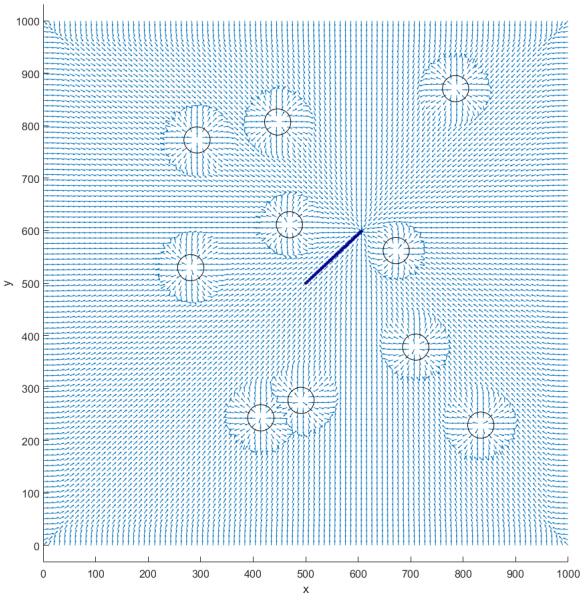


Figure 11: Obstacle created in the CoppeliaSim scene

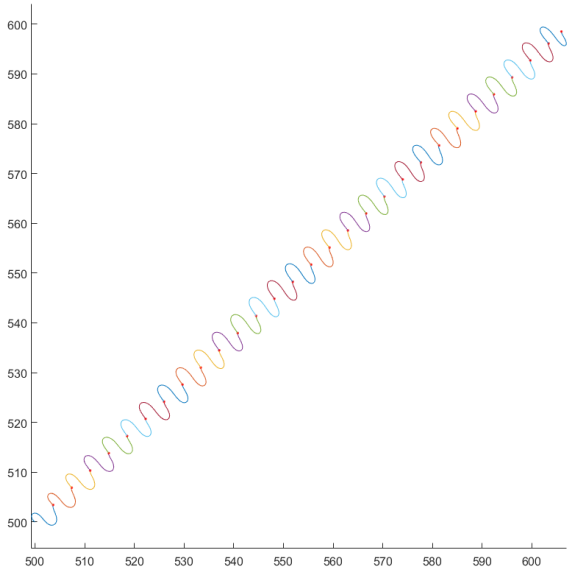
It is possible to see red body of the obstacle and black circle representing its influence field.
To steer the robot in simulator, values of desired angle of displacement of the arms are computed in each time instant, on the basis of assumed form of Fourier trigonometric series and values of coefficients obtained with MATLAB.
Due to fact that successive coefficients were stored in the array, it was possible to move the robot and occasionally read arriving coefficients. This approach allowed for undisturbed movement of the robot on long paths.

4 Results of conducted experiments

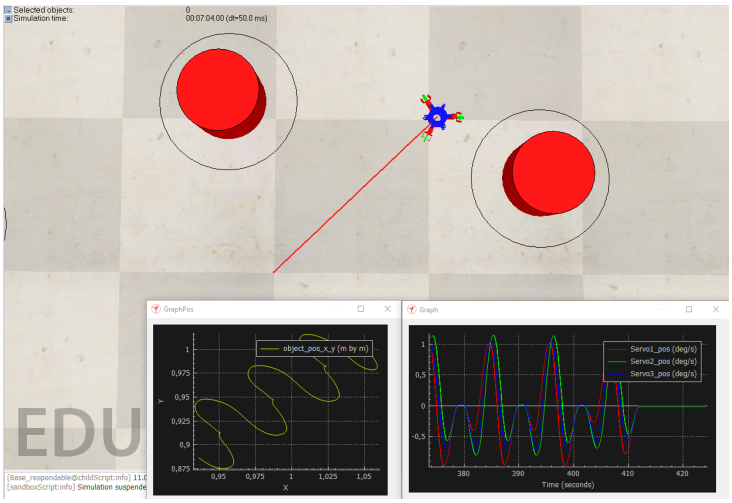
Sample simulation results are presented below. In both cases, robot has moved to the desired position with sufficient accuracy and avoided collisions with the obstacles.



(a) Desired path in gradient field

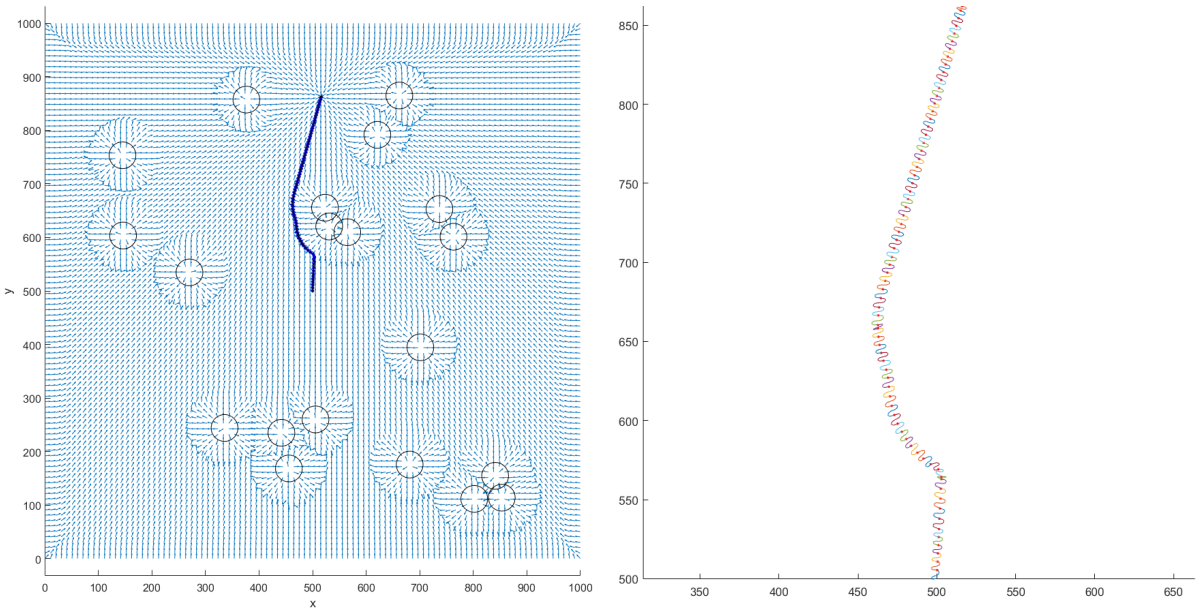


(b) Computed movement



(c) Path traveled by model of the robot in the simulator

Figure 12: Movement of robot in simple case (no obstacles on the robot path)



(a) Desired path in gradient field

(b) Computed movement

(c) Path traveled by model of the robot in the simulator

Figure 13: Movement of robot with obstacle avoidance

5 Summary

Implemented algorithm was working correctly. The results obtained with MATLAB and CoppeliaSim simulations were similar, which could mean that presented solution of obstacle avoidance is possible to be used on a real robot. Additionally, time taken for the algorithm to produce valid paths, even for complicated cases, was admissible.
Unfortunately, few problems have been encountered during experiments:

- the value of parameter responsible of size of influence of obstacles, if set incorrectly, allowed for collision of robot with obstacles,
- the potential field method in some cases generated local minima, for which the algorithm was producing incorrect paths.

It would be possible to use different high-level pathfinding algorithm instead of potential field method as to circumvent these issues.
As a final conclusion, the goal of this project has been accomplished.

References

[1] Mcmaster-carr website. https://www.mcmaster.com/. (access 01.02.2022).

[2] Masato Ishikawa. Trident snake robot: Locomotion analysis and control. *IFAC Proceedings Volumes*, 37(13):895 – 900, 2004. (6th IFAC Symposium on Nonlinear Control Systems 2004 (NOLCOS 2004), Stuttgart, Germany, 1-3 September, 2004).

[3] Masato Ishikawa, Yuki Minami, and Toshiharu Sugie. Development and control experiment of the trident snake robot. *IEEE/ASME Transactions on Mechatronics*, 15(1):9–16, February 2010.

[4] Janusz Jakubiak and Krzysztof Tchoń. Endogenous configuration space approach to mobile manipulators: A derivation and performance assessment of Jacobian inverse kinematics algorithms. *International Journal of Control*, 76(14):1387–1419, September 2003.

[5] Zuzanna Pietrowska and Krzysztof Tchoń. Dynamics and Motion Planning of Trident Snake Robot. *Journal of Intelligent & Robotic Systems*, 75(1):17–28, July 2014.

[6] Elia Nadira Sabudin, Rosli Bin Omar, and Che Ku Nor Azie Hailma. Potential field methods and their inherent approaches for path planning. *ARPN Journal of Engineering and Applied Sciences*, 11(18):10801–10805, September 2016.

[7] Coppeliasim remote api guide. https://www.coppeliarobotics.com/helpFiles/en/remoteApiClientSide.htm. (access 01.02.2022).

[8] Urdf file specification. http://wiki.ros.org/urdf/XML. (access 01.02.2022).