BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Robotics and AI

# MPC-MAP Assignment No. 3 - Report

**Author:** Tomáš Frigyik
**Date:** 10.3.2025

## Task 1 – Prediction

Discussion: Prediction function predict_pose was implemented, it returns the new pose, to new pose is added random noise by funcion.

## Task 2 – Correction

Discussion: Function compute_lidar_measurement function was implemented, it returns vector of measured distanes without any noise. Advantage of the function ray_cast has been taken.

Also weight_particles weighting function was implemented, for weighting was used formula 1.

$$w \propto \prod_{m=1}^{M} e^{-\frac{1}{2}\left(\frac{d_m - p_m}{\sigma}\right)^2} \tag{1}$$

## Task 3 – Resampling

Discussion: Resampling function resample_particles was implemented, it’s returning a new set of resampled particles. The implementation uses the systematic resampling algorithm.

## Task 4 – Localisation

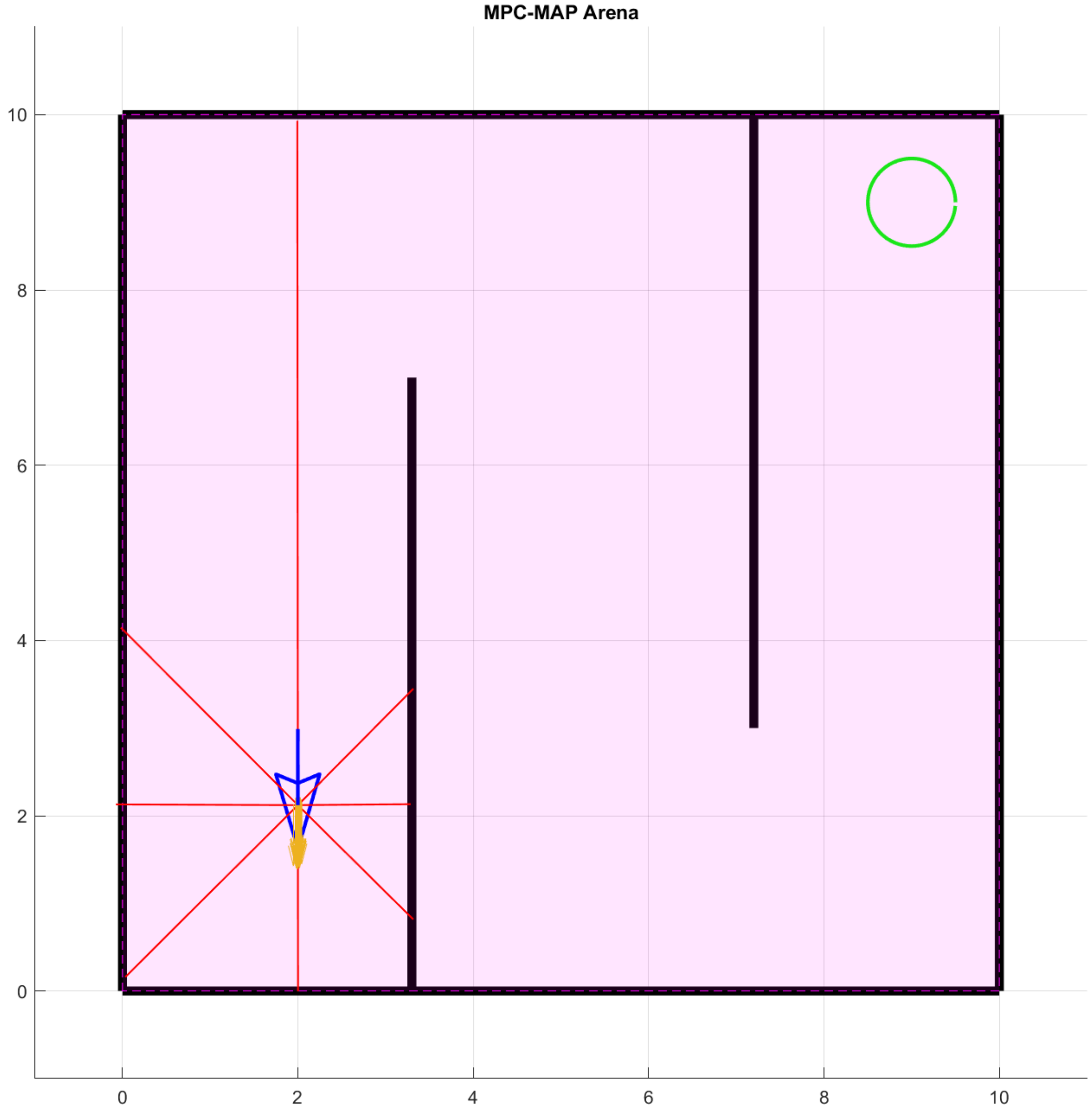

*Figure 1 - Cluster of particles gathered around the agent*

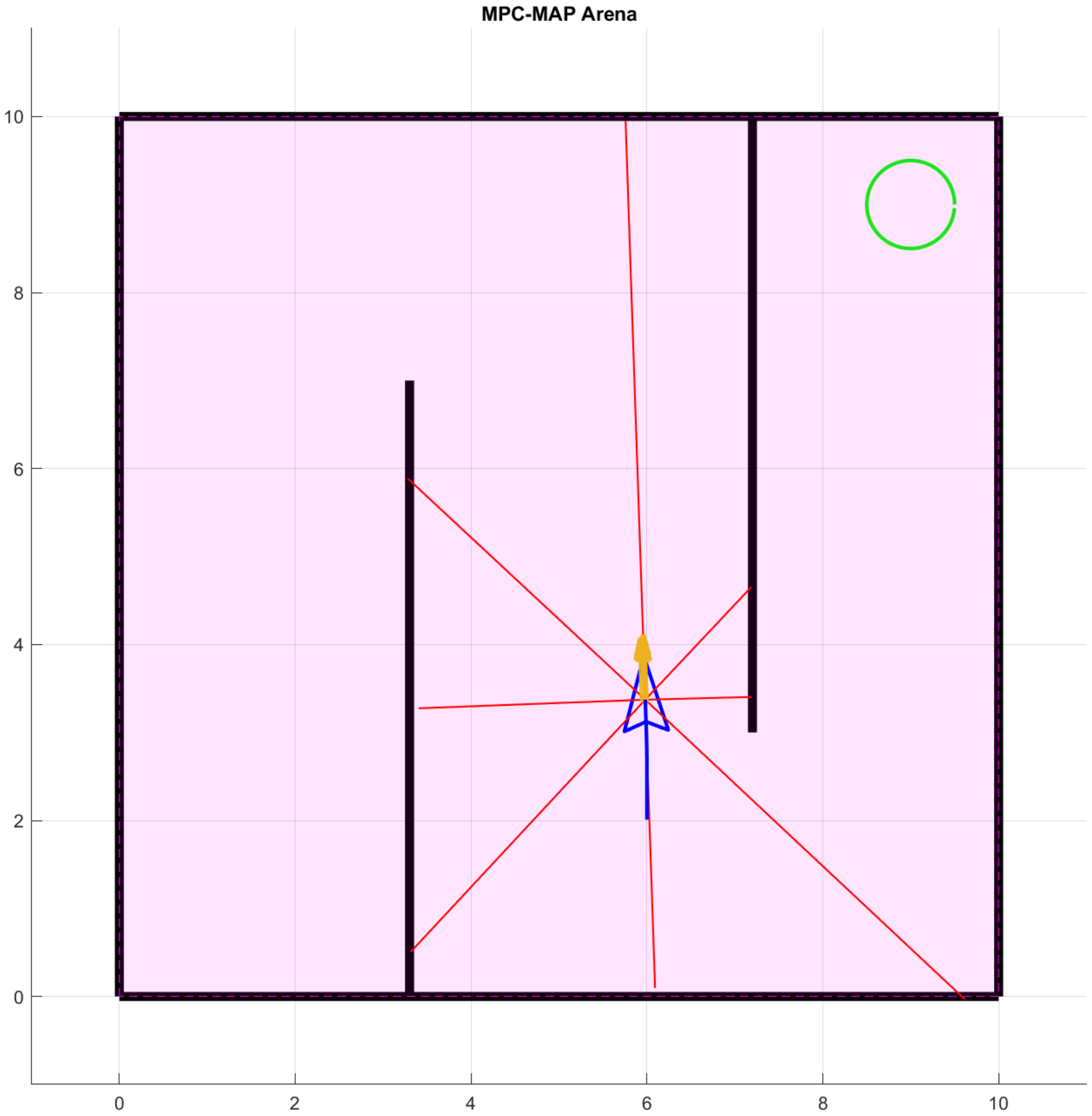

*Figure 2 - Cluster of particles gathered around the agent*

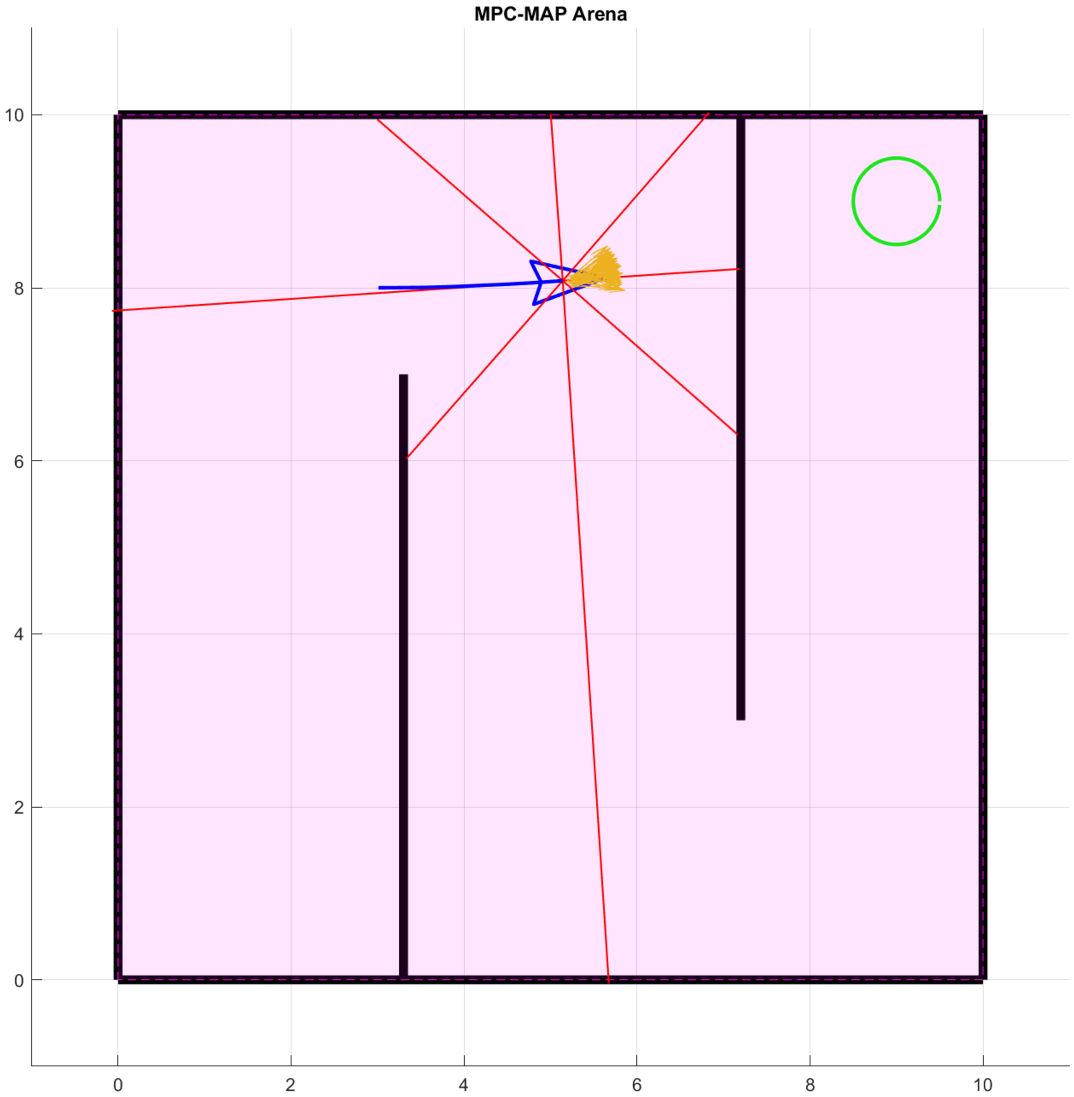

*Figure 3 - Cluster of particles gathered around the agent*

Discussion: Most important parameters of my solution are number or patricles, more particles generated at the beginning of the algorithm leads to faster convergence, and also, if there is higher noise implemented for new pose of the particle, the algorithm converges as later, as bigger noise is implemented.

Because I was implementing differential drive, for prediction was used equation for differential drive with implementation of noise for simulating real world conditions. For correction was used Gaussian probability function, becasue thanks to that, I could determine which particles are important for localisation, weight each particle and use them in next iteration of the algorithm. For resampling was used Systematic algorithm, because it’s commonly used method in particle filters, it’s efficient and accurate.

The major issue I have had to overcome was to find out how to work with intersections of the lidar rays and map walls and with implementing prediction of the robot pose based on variable interwheel_dist.