ROBT 403: Robotics II

Homework III

Madi Nurmanov
Nazarbayev University
Astana, Kazakhstan
madi.nurmanov@nu.edu.kz

***Abstract*—This homework is based on the implementation of differential drive robot forward kinematics using simple artificial neural network. Inputs are angular velocities of the wheels and goal position, where the light source is placed.**

Keywords—differential drive robot; forward kinematics; ANN

I. Introduction

There are different techniques for definition of the position and orientation of the robots in spatiotemporal sequence. It is useful for navigation, path planning and localization.

For differential drive robots, the forward kinematics is developed in reliance on the principle of Instantaneous Centers of Curvature. As the robots have two independent motors, the robot's movement, when the wheels are rotating in the same direction with different speeds, is a circular curvature with radius equal to the difference between this Instantaneous Center of Curvature and the central point P of the axis between the wheels.

Considering this fact, it is possible to derive equations at which wheel are moving with respect to the ICC:

$$\omega(R+l/2) = V_r \qquad [1]$$
$$\omega(R-l/2) = V_l \qquad [2]$$

And the position of the central point P in time is defined by the following set of equations:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\,\delta t) & -\sin(\omega\,\delta t) & 0 \\ \sin(\omega\,\delta t) & \cos(\omega\,\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\,\delta t \end{bmatrix} \qquad [3]$$

Whereas, coordinates of ICC are obtained dynamically at each iteration from the coordinate difference:

$$ICC = [x - R\sin(\theta),\ y + R\cos(\theta)]$$

This report is based on several approaches, one of which is different, in the sense that it doesn't require the definition of ICC. Robot dynamics is being simulated in the Matlab environment: application builder.

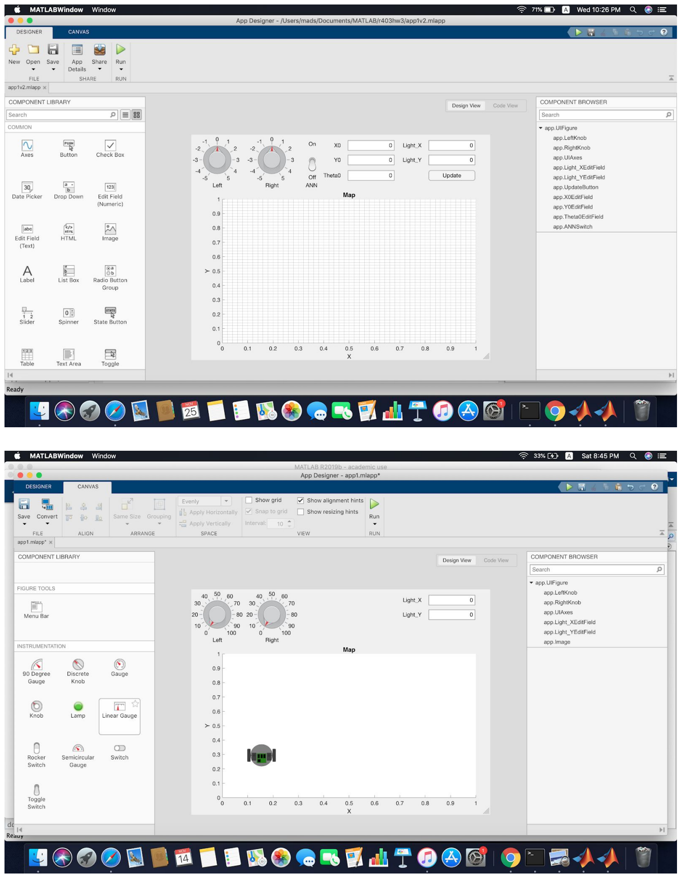


figure 0: (a) Application builder interface in MatLab

(b) Using DDR picture as a model

II. Implementation

Part I. hgtransform approach

To run the simulation simply open the app1.mlapp file, run it from the MatLab.

In the new window with the GUI change the values for position of the light source and starting position and orientation of the robot, at will. Then push the "UPDATE" button.

IMPORTANT: DO NOT TOGGLE SWITCH ON, AS IT WILL ACTIVATE THE ARTIFICIAL NEURAL NETWORK.

In this implementation, the inputs are taken as knob values that define angular velocities of left and right wheel respectively.

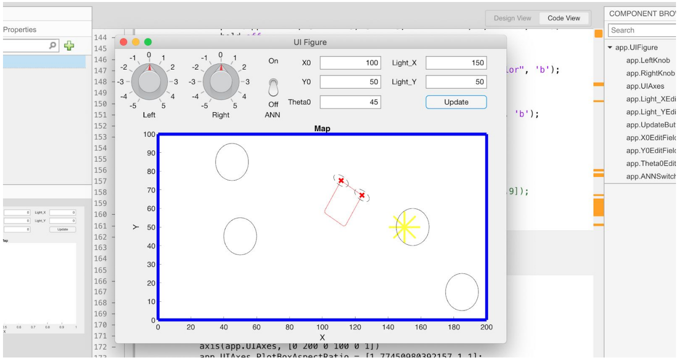


figure 1: ICC independent kinematics for DDR

This approach gives robot basic functionality; it rotates around itself when wheels are rotating in opposite directions,

If only one wheel is stationary, the robot rotates around it and if the velocities are in the same direction, the robot moves forward.

It should be noted that obstacle avoidance is not realized in this approach. The reason for that is the fact that hgtransform function of Matlab, which is the main tool of this implementation for rotation and translation of robot with respect to its central point p is dynamic. It means that the distance or “luminosity” received by the sensors is inconclusive, as the frame of the robot is being moved around the map, internally, the position of the sensors and the robot itself with its base frame are stationary. However, it's the only one useful for the resemblance oriented implementations, as curved rectangles and circles can be generated in MatLab only as a separate type of object, different from, for instance, plotting rectangle as a set of points. In addition to that, animation in the MatLab application builder is realizable only using this function.

Part II. Forward Kinematics using ICC

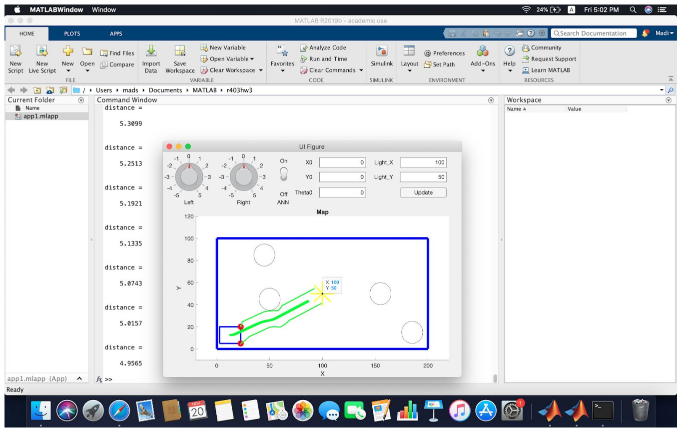
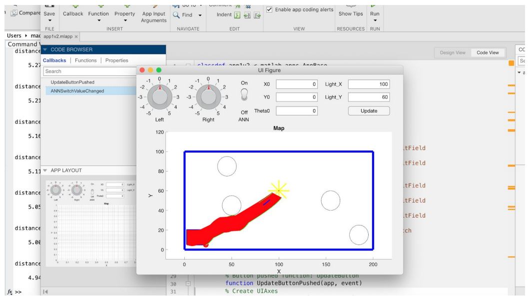
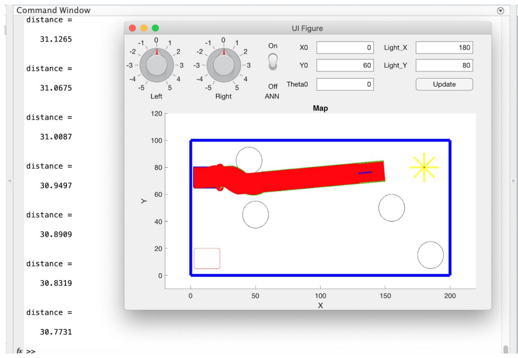


figure 2: Path planning for DDR using ANN

In this implementation, the predefined ICC dependent kinematics model was used. The model relies on the equations of instantaneous position and orientation of the robot.





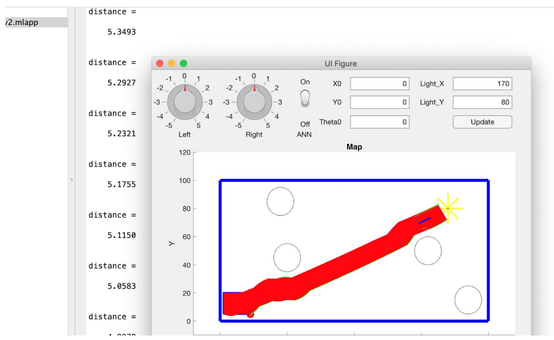


figure 3: (a)(b)(c) Robot path from different start point to the light source.

The red path is the position of the robot after each iteration, while the green points are positions of sensors. It is important to keep them, as otherwise only final positions will be shown, which questions the quality of the ANN.

III. Discussion

As it can be seen, both hgtransfrom and ICC models of differential drive robots have their limitations in MatLAB application builder. As an implication for the future, it is advised to simply use matlab scripts, python or Mobile Robotics Simulation Toolbox in Matlab, as they show better performance. However, the algorithm for ANN based control of DDR is fully realized using the Matlab Application builder environment.