Code Documentation for the Robot Navigation Simulator

Acknowledgment

The original code documentation was done by the initial contributor Michal Kramarczyk. This is a revised version of the code documentation that describes the original features and new features we introduced to the simulator.

Information

- Read Section 1 to learn how to run the code
- Read Section 3 to learn about parameters that can be modified
- Read other sections for other functions & scripts in the program

1. Running the code

1. For running the code, use ***Main.m*** file.
2. Run each subsection **separately** using CTRL+ENTER key combination.
3. First, run ***Subsection I*** to initialize all variables.
4. Run ***Subsection II*** to gather path dataset. – Once the dataset is created, this dataset is saved as an external file and can be used again and again in training MLPs or RNNs. Refer to *save()* and *load()* functions on MATLAB.
5. ***Subsection Extra*** can be used to make a backup file for the dataset.
6. Run ***Subsection III – I*** or ***Subsection III – II*** to train the RNN or MLP. The trained network will be saved as an external file.
7. Run ***Subsection III – I*** or ***Subsection III – II*** to test the RNN or MLP.

2. Files in *functions* folder

Most functions are used throughout the simulator systems. Most of them need not be modified unless you want to modify the simulator itself. Some functions are utilities that can be called manually.

- ***DrawMaze.m*** – (Utility) visualizes a maze with goals.

  Usage: DrawMaze(maze_files, mazeGoal, 'ID of a maze' (ex: 1 for maze01))
- ***Drive.m*** – (Function) drives the robot for designated distance with specified wheel configurations.
- ***GenerateMaze.m*** – (Function) loads a maze labeled in *maze_files*.
- ***GenerateRandGoal.m*** – (Function) generates a random goal instead of a fixed goal.
- ***InitialLaserRead.m*** – (Function) creates initial sensor readings at the initial position and orientation.
- ***KinUpdate.m*** – (Function) updates the position and orientation of the robot
- ***LaserMeas.m*** – (Function) measures the distances to the closest obstacles in each sensor's direction.
- ***LineCross.m*** – (Function) returns the coordinate where two lines cross
- ***P2L.m*** – (Function) refer to the comments in the file
- ***ReverseKin.m*** – (Function) reverse kinematics. Transforms the robot's linear and angular velocity to wheels' velocity.
- ***SaveFigure.m*** – (Function) saves the path navigation result to an external file. Refer to the comments in the file
- ***Simulation.m*** – (Utility) simulates the path navigation. Refer to a usage in ***main.m***.
- ***V2L.m*** – (Function) refer to the comments in the file

3. Files in *inits* folder

Four MATLAB scripts are for initializations of neural networks and goals. **Most parameters of concern in training the neural networks are initialized in these scripts.**

- ***InitMLP.m*** – Initializes parameters used for MLP.
  1. ***mlpParam*** – A structure for MLP parameters
     - ***moment*** – Momentum in weight updates
     - ***alpha*** – Slope scale of sigmoid function
     - ***eta*** – Leaning rate
     - ***epoch*** – Number of Iterations
  2. ***mlpArch*** – A structure defining MLP network structure
     - ***inputs*** – Number of nodes in input layer
     - ***hidden1*** – Number of nodes in 1st hidden layer
     - ***hidden2*** – Number of nodes in 2nd hidden layer
     - ***outputs*** – Number of nodes in output layer
  3. ***wIn1Init*** – Initial weights connecting input layer and 1st hidden layer
  4. ***w12Init*** – Initial weights connecting 1st hidden layer and 2nd hidden layer
  5. ***w2OutInit*** – Initial weights connecting 2nd hidden layer and output layer

- ***InitRNN.m*** – Initializes parameters used for RNN.
  1. ***nn*** – A structure for RNN
  2. ***nn.option*** – A substructure for RNN parameters
     - ***subset_length*** – Length of path segment used per path data
     - ***learningRate*** – Learning rate
     - ***maxIter*** – Number of Iterations
     - ***numHidden*** – Number of nodes in the hidden layer recurrency
     - ***numInput*** – Number of nodes in input layer
     - ***numOutput*** – Number of nodes in output layer
     - ***netDim*** – Net RNN structure. Refer to the comments in the file.
     - ***activation* –** Activation function
     - ***dActivation*** – Derivative of Activation function
  3. ***nn.layer*** – A substructure defining RNN structure. Weights and biases are stored in here. Initialized by ***initializeNN*** function in ***RNN*** folder.

- ***InitRobot.m*** – Initializes parameters that define the robot.

  1. ***sensor_num*** – Number of obstacle sensors
  2. ***sensor_ang*** – Max. obstacle sensor direction. Refer to the comments in the file.
  3. ***robot*** – A structure defining robot's status. Refer to the comments in the file.
     - ***pose*** – Robot's current position and orientation
     - ***param*** – Robot wheel settings
     - ***size*** – Robot's shape
     - ***laserAngles*** – Directions of obstacle sensors
     - ***goal*** – Target goal
  4. ***collision*** – Flag for collision
  5. ***goal*** – Flag for reaching the goal
  6. ***poseHist, laserHist, velHist, gtHist*** – Containers for data collection

- ***loadMazeGoal.m*** – Initializes some designated goals for maze files.

  1. ***mazeGoal*** – Goals for mazes in ***mazeLib***
  2. ***mazeDumpGoal*** – Goals for mazes in ***mazeDump***. Some are impossible to reach.

4. Files in *MLP* folder

The functions are for training and testing the multilayer perceptron. **Structural changes should be made in this folder.**

- ***MlpRun.m*** – Push a sample into the trained MLP to get the output
- ***MlpTrain.m*** – Train the MLP using the dataset
- ***sig.m*** – Sigmoid function
- ***sigDer.m*** – Derivative of sigmoid function

5. Files in *RNN* folder

The functions are for training and testing the **Elman-type** recurrent neural network. **Structural changes should be made in this folder.**

- ***activation*** folder – Folder containing activation functions
    1. ***dSigmoid.m*** – Derivative of sigmoid function
    2. ***sigmoid.m*** – Sigmoid function

- ***BPTT.m*** – Backpropagation through time function
- ***FFTT.m*** – Forward propagation function
- ***initializeNN.m*** – Initializes the weights and biases of all layers
- ***nnSingleFF.m*** – Push a sample into the trained RNN to get the output
- ***trainRNN.m*** – Train the Elman-type RNN using the dataset

6. Files in *scripts* folder

Four MATLAB scripts are for the automatic robot navigation used for collecting the training dataset. **Based on the robot laser sensor configuration, these scripts must be modified accordingly.**

- ***HistoryUpdate.m*** – Updates history for data collection.
- ***Nav2Goal.m*** – A navigation script for reaching a goal. Acts differently based on whether the goal is currently visible or not.
- ***StaticRoute.m*** – (Outdated) a navigation script that just follows a designated path.
- ***TunnelDrive.m*** – (Outdated) a navigation script that just avoids collision in narrow passage.
- ***WallFollow.m*** – (Outdated) a navigation script that just follows a wall to avoid collision.

7. Role of other folders

- ***data*** – The folder where collected dataset, trained networks, and simulation results would go. **All folders in *data* folder should NOT be deleted** (yet their contents may be deleted). They will not be automatically created when running the code. The files initially in this folder are just sample dataset and trained networks from our experiment. They will be overwritten when you run the code.
- ***mazeDump*** – The folder for maze files that are not used. Most of these mazes contain some impossibilities or flaws that the robot cannot navigate through them. You can modify these mazes and put into ***mazeLib*** folder to use more mazes in training/testing.
- ***mazeLib*** – The folder for maze files used in training/testing. By default all maze files in this folder are used for path data collection.
- ***mazeTemplate*** – The folder containing an empty maze template. You can draw a new maze using the maze file in the folder.