User Manual

Toyota - AGV

December 2021

Version 1.0

Status

Reviewed	Björk, Rasmus	2021-11-26
Approved		

Project Identity

Group E-mail:	carhe007@student.liu.se
Homepage:	https://tsrt10.gitlab-pages.liu.se/2021/toyota/
Orderer:	Anton Kullberg Phone: Not applicable E-mail: anton.kullberg@liu.se
Customer:	Oskar Bergkvist (Toyota Material Handling) Phone: Not applicable E-mail: oskar.bergkvist@toyota-industries.eu
Supervisor:	Hamed Haghshenas Phone: Not applicable E-mail: hamed.haghshenas@liu.se
Course Responsible:	Daniel Axehill Phone: Not applicable E-mail: daniel.axehill@liu.se

Project members

Name	Post	E-mail
Carl-Hampus Hedén	Project manager	carhe007@student.liu.se
Mahdi Najafi	-	mahna987@student.liu.se
Alfed Boman	Head of design	alfbo741@student.liu.se
Adam Kagebeck	-	adaka206@student.liu.se
Kalle Blomkvist	Head of software	karla625@student.liu.se
Rasmus Björk	Head of documentation	rasbj268@student.liu.se
Viktor Ekström	Head of testing	vikek514@student.liu.se

CONTENTS

DOCUMENT HISTORY

Version	Date	Changes	Made By	Reviewed
0.1	2021-11-26	Draft	All	Rasmus Björk
1.0	2021-12-14	Final version	All	Mahdi Najafi

1 INTRODUCTION

The user manual describes all that the user must know in order to start using the system. The user manual will show how the user interface looks, how the user interface is used, and all the functions available to the user. It is highly recommended to read this document before proceeding with using the system.

2 SYSTEM REQUIREMENTS

Below is a specifications of the performance requirements for running the system.

2.1 Software

To use the program there are some software that needs to be installed. To run the program a version of MATLAB 2021a and simulink version 10.3 is required together with the following toolboxes.

- MATLAB, version 2021a
- Simulink, version 10.3
- Deep Learning Toolbox, version 14.2
- Optimization Toolbox, version 9.1
- Reinforcement Learning Toolbox, version 2.0
- Statistics and Machine Learning Toolbox, version 12.1

In addition to this, python 3.8 is required to be installed together with the following packages:

- matplotlib, version 3.3.3
- numpy, version 1.17.4
- scipy, version 1.6.0

Other versions of the listed software might also be compatible with the implemented system, but they have not been tested.

2.2 Hardware

The hardware requirements consists of a computer that is able to run MATLAB. Any Intel or AMD x86-64 processor should work but en increase in performance will speed up the simulations. It is recommended to have at least 16 GB RAM.

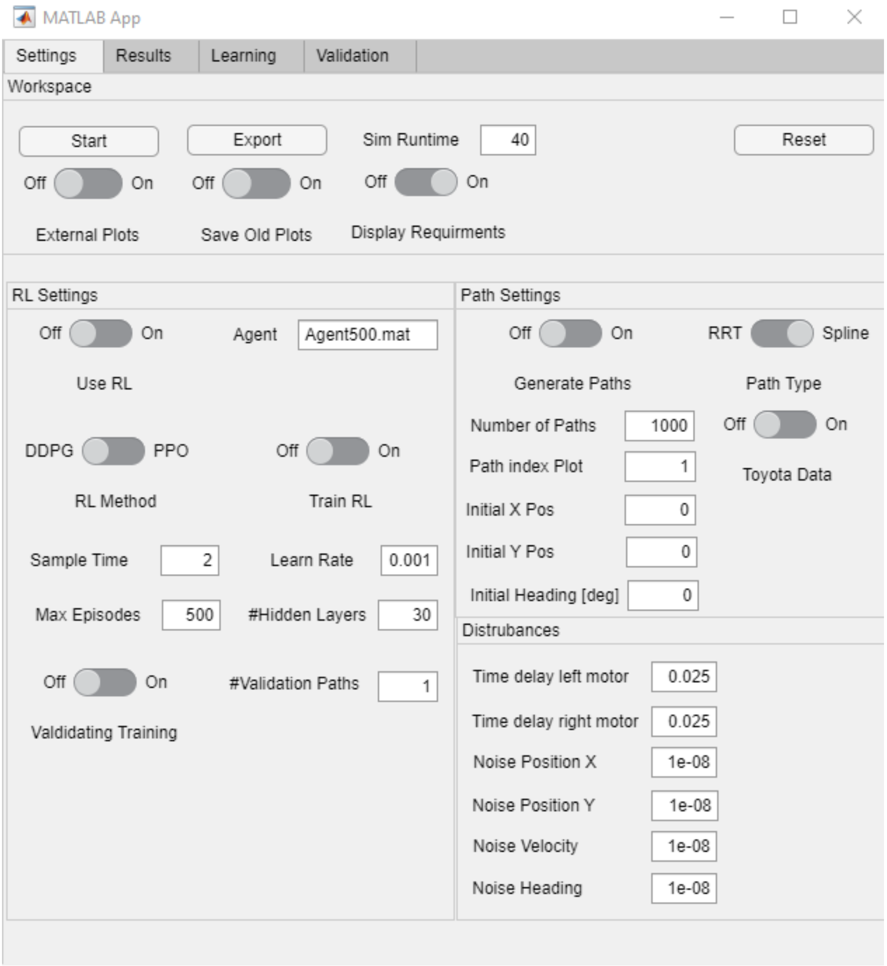
- Processor: Any Intel or AMD x86-64 processor with four logical cores and AVX2 instruction set support.
- Memory: 16 GB
- Disk space: 15 GB
- Graphics: No specific graphics card is required.

3 RUNNING THE SYSTEM

In order to initialize the system, open the GUI and enter the desired settings. The settings are described in section 3.1. In order to run the system, click the *Start* button in the top left corner in the settings tab.

3.1 Settings tab

The settings tab can be viewed in Figure 1.



Sim Runtime 40

Off On External Plots

Off On Save Old Plots

Off On Display Requirments

Off On Use RL

Agent Agent500.mat

DDPG PPO RL Method

Off On Train RL

Sample Time 2

Learn Rate 0.001

Max Episodes 500

#Hidden Layers 30

Off On Valdidating Training

#Validation Paths 1

Off On Generate Paths

RRT Spline Path Type

Number of Paths 1000

Path index Plot 1

Initial X Pos 0

Initial Y Pos 0

Initial Heading [deg] 0

Off On Toyota Data

Time delay left motor 0.025

Time delay right motor 0.025

Noise Position X 1e-08

Noise Position Y 1e-08

Noise Velocity 1e-08

Noise Heading 1e-08

Figure 1: Settings tab of the GUI

3.1.1 *Workspace panel*

- Start: Button to run the program.

- Export: Button to save setting variables to the work space. The start button does this automatically, mainly used is for debugging.
- Sim Runtime: Indicate how long the each simulation should be in seconds.
- External plots: Enable to have plots automatically appear in external windows separate from the program.
- Save Old Plots: If enabled the program will not overwrite the plots from the previous runs. If disabled only the current plots will be opened. Note that this setting only relates to external plots.
- Display Requirements: Setting to for showing lines for requirement 26 and 27.
- Reset: Clear all variables in the work space and close all plots.

3.1.2 *RL Settings: Settings for reinforcement learning*

- Use RL: On/Off switch for running with our without machine learning agent. If turned off only base controller is used.
- Agent: Name of the file of the previously trained agent to be used if *Train RL* is disabled. Note that the agent should be saved in the folder called *Saved Agents*
- Train RL: On/Off switch indicate if the system should train a new agent or use a preexisting. If there is no pretrained agent and the system uses an untrained agent if this switch is turned off.
- RL Method: Switch to decide what reinforcement algorithm should be used, PPO, or DDPG.
- Sample time: Value used for sample time in seconds. This corresponds to how often the agent should output an action. The more actions that wants to be received in an episode the lower this value should be. Since lower sample time means more actions this will slow down the training. Suitable values can be in the range of 1 to 5.
- Max episodes: Value of episodes that agent will maximally train.
- Learn rate: Enter the learn rate used for the agents training. The parameters has a value between 0 and 1 and affects how the agent weighs the new vs. the old value. For a stable and reliable, however slower, training the parameters should be set to a small value e.g. 10^{-5} to 10^{-3}. In this project the same rate is used for both the actor and the critic.
- #Hidden Layers: Number of parameters in the hidden layers used when training the agent. A high value gives a larger, more complex, network while a smaller value yields the opposite. Larger networks needs more time to run and can risk overfitting. A small network is faster but can have harder to solve the problem at question. For this problem a suitable value can be between 15 and 50.
- Validated training: Tell the system whether it should run validation test on the reinforcement learning agent.
- #Validation Paths: Number of paths that should be used in validating the agent.

3.1.3 *Path Settings*

- Generate Paths: Tells the system if it should generate a new list of paths.
- Path Type: Tells the system what method paths should be generated by, RRT or Spline. Note that there are preexisting paths in the work space, therefore this option is not dependent on the setting of Generate Paths.
- Number of paths: Enter the value of paths that the agent should use for training. Value must be an integer larger than zero.
- Path plot index: Tell the system what index from the path list that should be visualized. This value must be larger than 0 and smaller than *Number of paths*. Note that if you want to visualize several paths you will need to run the program multiple times, however if *Generate Paths* is turned on the program will randomize new paths between runs.
- Initial X Pos: Set the initial value for the X position of the AGV.
- Initial Y Pos: Set the initial value for the Y position of the AGV.
- Initial Heading: Set the initial heading of the AGV in degrees.
- Toyota data: Tells the system if it should use Toyota data. This is a setting for validating the kinetic model the simulator is based on. If you want to switch the data file used in this validation you need to:
  1. Open the `Toyota.prj` file.
  2. Navigate to the `Toyota_ref_u.slx` file located in the Componentes/Controller folder.
  3. Open the *from spreadsheet* block and select the desired file. Note make sure the correct column data is plugged into the system and that the first column indicates run time for the system.

3.1.4 *Disturbances settings*

- Time delay left motor: Indicate the how long the time delay from the left electric motor should be in seconds.
- Time delay right motor: Indicate the how long the time delay from the right electric motor should be in seconds.
- Noise Position X: Indicate the power of the measurement noise of the AGV's position in the X direction. The noise is a continuous uniform distribution between zero and the selected value.
- Noise Position Y: Indicate the power of the measurement noise of the AGV's position in the Y direction. The noise is a continuous uniform distribution between zero and the selected value.
- Noise Velocity: Indicate the power of the measurement noise of the AGV's current velocity. The noise is a continuous uniform distribution between zero and the selected value.
- Noise Heading: Indicate the power of the measurement noise of the AGV's current velocity. The noise is a continuous uniform distribution between zero and the selected value.

4 SYSTEM OUTPUT

Below is a description on how the user can access the results/outputs of the system.

4.1 Results tab

The results tab can be viewed in Figure 2.

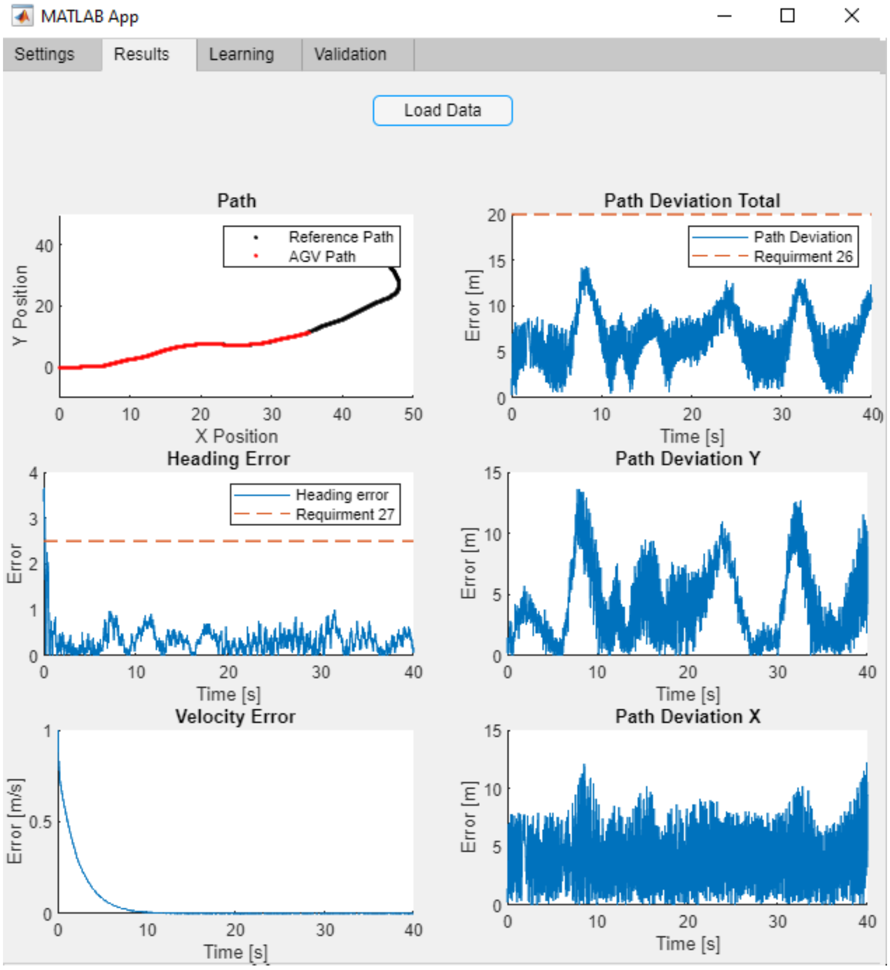


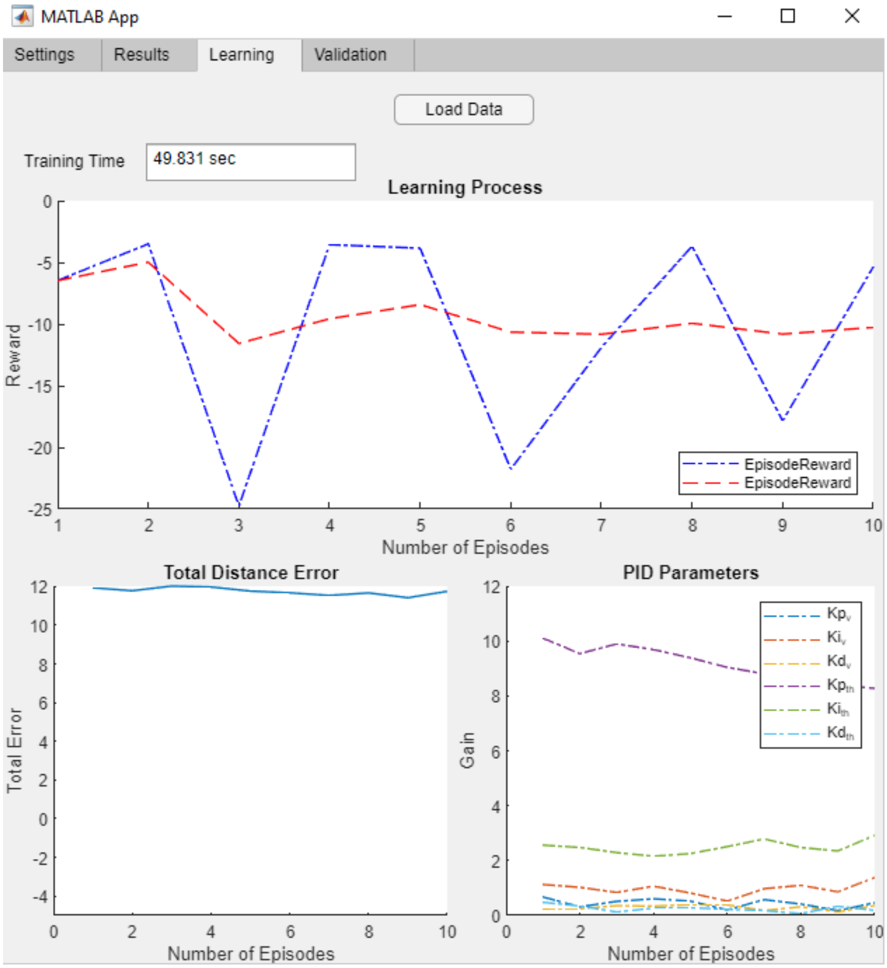
Figure 2: Results tab in the GUI

- Load data: Button to load data to the charts bellow the button. Data is from the path with the index assigned at *Path plot index* (described in 3.1.3) the previous run.
- Path: The reference path is indicated in black, and the path followed by the AGV is indicated in red. X,Y axes indicate position.
- Heading Error: Indicate how the heading error changes over time.
- Velocity Error: Indicate the size of the velocity error charges over time.
- Path Deviation Total: Indicates how the total distance to the path over time.

- Path Deviation X: Indicates the distance to the path in the x direction over time.
- Path Deviation Y: Indicates the distance to the path in the Y direction over time.

4.2 Learning tab

In this section the training data for the agent is visualized and the learning tab can be viewed in Figure 3.



Training Time: 49.831 sec

Figure 3: Learning tab in the GUI

- Load data: Button to load the training data to the charts bellow the button.
- Training Time: Displays the time the agent took to train.
- Learning Process: Chart visualizing the reward for each episode (blue) as well as a rolling average for the previous 100 episodes (red). The number of episodes in the rolling average can be changed via the *ddpg.m* and *PPO.m* files respectively in the *ScoreAveragingWindowLength* setting.

- Total Distance Error: Chart displaying a summation of the distance error over the trained episode.
- PID Parameters: Chart displaying the average value for each of the PID parameters over the trained episodes.

4.3 Validation tab

The validation tab can be viewed in Figure 4.

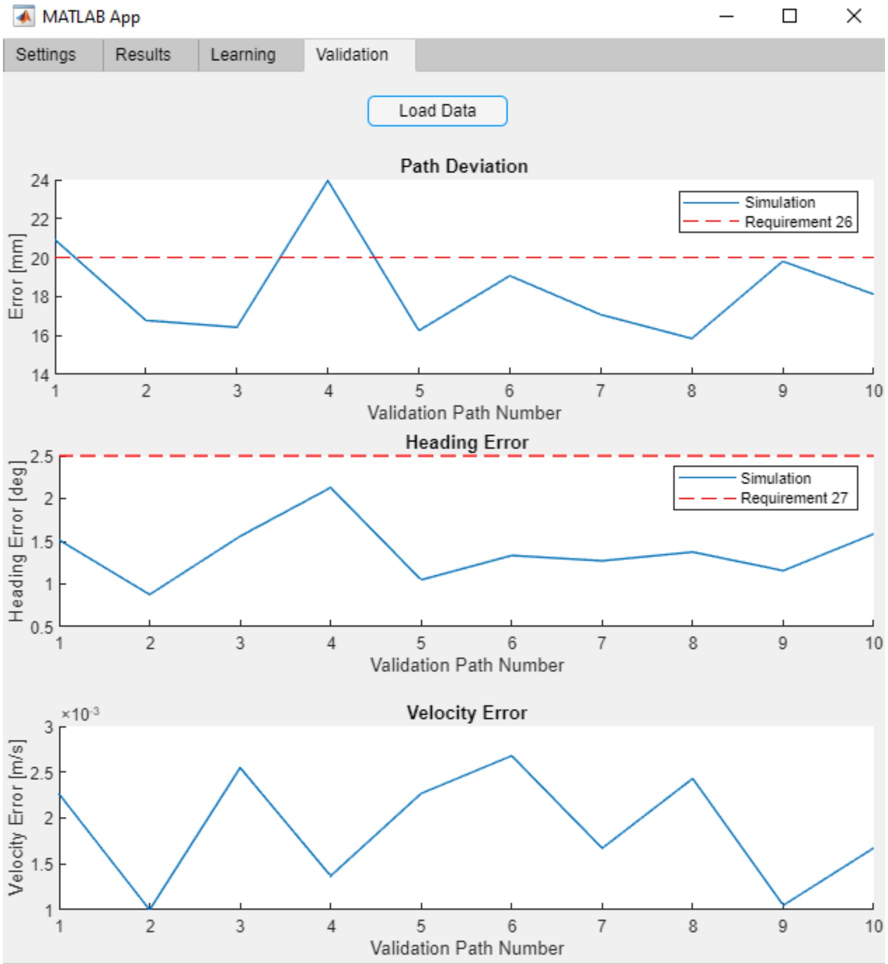


Figure 4: Validation tab in the GUI

- Load data: Button to load the validation data to the charts bellow the button.
- Path Deviation: Chart displaying the max distance error for each of the validation paths. Note that the initial stabilization values have been clean from this data.
- Heading Error: Chart displaying the max heading error in degrees for each of the validation paths. Note that the initial stabilization values have been clean from this data.

- Velocity Error: Chart displaying the max velocity error in degrees for each of the validation paths.Note that the initial stabilization values have been clean from this data.