**ReadMe for Software of Time Optimal trajectories for a car-like robot**

1. **Program**
   a. **Description:** This source code program can produce a minimum-length trajectory for any two configurations.
   b. **Size:** 28kb
   c. **Platform:** Matlab 7.1.
   d. **Major Component Description:** 41 .m files in the folder `Program`
      Three main entry functions are in:
      1. `OptimalTrajectoryGUI.m` for producing the Graphical User Interface (GUI) which demonstrating the optimal trajectories and the corresponding geometric representation.
      2. `MinimumLengthTrajectory.m` for producing the time-optimal trajectory from [0,0,0] to any specified target configurations
      3. `OptTrajTwoConf.m` for producing the time-optimal trajectory from [0,0,0] to any specified target configurations.
   e. **Detailed Run Instructions for GUI:**
      1. In Matlab Command Window: Type ‘OptimalTrajectoryGUI’ and Enter, you will see a GUI; Or open `OptimalTrajectoryGUI.m` then click run.
      2. Input a final configuration as a vector [x, y, \( \theta \)], where [x, y] is the position of the desired final configuration of the car, whereas \( \theta \) is its final orientation with respect to the x-axis. Note that, x, y and \( \theta \) can be any real numbers.
      3. Click ‘Show’ button, a) you can get a minimum-length trajectory to link the initial configuration (where we assume it is [0,0,0]) to the final configuration in the below figure; b) The optimal control law shows in the right upper text window and its corresponding elapsed time; c) the right bottom figure shows its corresponding start and final position of \( \omega_t \). The blue one is for the start \( \omega_t \) and the red one is for the final \( \omega_t \).
      4. Click ‘Run’ button, then a) the optimal control law is shown in the right upper text window; b) a car is running along the optimal trajectory in the left bottom figure, c) Correspondingly, the rotation of its \( \omega_t \) shows in right bottom figure.
   f. **For further use:**
      The GUI is used to demonstrate the results and concepts used in the paper.
      If readers want to use its result in their research for determining the minimum-length or time-optimal trajectory, just call function:
      ```
      Trajectory=MinimumLengthTrajectory(finalConfiguration).
      ```
      1) **Input:** Give the finalConfiguration as a 1x3 vector such as [1, 1.01, 1].
      2) **Output:** Trajectory has two fields. Trajectory.mod is 2x\( n \) matrix, first row is for control \( u \) and second one for control \( v \), and corresponding elapsed time in the same column in Trajectory.length which is 1x\( n \) vector.
      Or
      ```
      Trajectory= OptTrajTwoConf(startConfig, finalConfig)
      ```
      1) For example: Trajectory=OptTrajTwoConf([3 2 1],[1 2 3]).

2. **Contact Information**
   If you want to know more about algorithm and its grounded reasoning, please read the papers.
If you want to use the program in your research, please refer to the papers.

Any comment, advice or discussion is greatly welcome.

Please email me:

elizabethhf@gmail.com (Huifang Wang)

Huifang Wang
Nov. 28, 2007