ECSE 683 Assignment - 2 Report

Adharsh Mahesh Kumaar

McGill ID: 260905451

1 Examples

Kinova Gen3 Ultralightweight robot arm is used to write 2D letters in this assignment as shown in Fig 1. The simulation is performed using MATLAB Robotic Tool Box.

Figure 1: Kinova Gen3

The configuration space of the Kinova Gen3 robot is defined by the position of all the joints which the robot can take. So the configuration space of the robot is R^7 and the degree of freedom (DOF) of the configuration space is 7 because the dimension of C-space is 7.

The task space is all the positions the Kinova Gen3 robot can occupy which is the 3D (x-y-z) plane so the degrees of freedom (DOF) is 3.

In learning from demonstration, our data is represented as $[\xi_1, \xi_2, ..., \xi_N]$.
Our state and action is given by:

$$\xi = \begin{bmatrix} \xi_n^s \\ \xi_n^a \end{bmatrix}$$

$\xi_n^s \;\epsilon\; R^P$ is the state of the trajectory and $\xi_n^a \;\epsilon\; R^Q$ is the action of the trajectory. Here our state ξ_n^s represents the input given to the model and our action ξ_n^a represents the output from the model. The input to the model is reference position. Therefore the state vector ξ_n^s is given by,

$$\xi_n^s = \begin{bmatrix} Y \\ Z \end{bmatrix}$$

where Y,Z represents the task space position needed by the robot to track the trajectory of the letter at each time step.

The output from the model or the action is the velocity predicted by our model.

$$\xi_n^a = \begin{bmatrix} V_y \\ V_z \end{bmatrix}$$

where V_y and V_z represents the predicted velocity from the Gaussian Mixture Model.

The state of the robot is the joint position of the robot and the action of the robot is the joint position calculated from inverse kinematics.

2 Method

The data set to perform the task can be found here. The data contains position, velocity, and acceleration required to track the trajectory at each time step. This data is interpolated to a spline function to form a trajectory. The last trajectory is used for testing and the remaining trajectories are used for training. Imitation learning is used to perform this task. The Expectation-Maximization (EM) method is used for training and Gaussian Mixture Regression (GMR) is used for prediction. The output of Gaussian Mixture Regression is the velocity needed at each time step to follow the trajectory of the letter.
This velocity is used to control the robot. We scale the velocity to limit our joint velocities by satisfying the kinematic constraints of the robot. By using inverse kinematics, we find the joint velocities of the Kinova Gen3 robot. The joint velocity is used to calculate the joint position at each time step and the calculated joint positions can be used to control the robot.

3 Code

The code is implemented in MATLAB using Matlab Robotic Tool Box.

Algorithm for Obstacle Avoidance:

- From the set of data points of each letter, we generate the trajectory using a spline interpolation function. After generating the trajectory, we use the last trajectory for testing and the remaining trajectories for training.

- Model is trained with a mixture of 5 Gaussians (model.nbVar = 5). Before learning, the model is initialized using the *initGMMkmeans* function [1].
- The Gaussian Mixture model is learned using the EM algorithm with the help of $EM - GMM$[2] function.
- The prediction is performed using Gaussian Mixture Regression (GMR) with the help of GMR function [3]. The output of Gaussian Mixture Regression is the velocity needed by the robot to track the trajectory.
- 3 initial configurations are given for the robot to write 3 letters. One letter is written at $[0, \pi/4, 0, \pi/4, 0, \pi/4, 0]$, The second letter is written at $[0, -\pi/3, 0, -\pi/3, 0, -\pi/3, 0]$ and the last letter is written at $[0, \pi/3, 0, \pi/3, 0, \pi/3, 0]$.
- The initial configuration and predicted velocity from our GMR model, is given to *drawC* function to perform inverse kinematics.

  The function *drawC* scales the predicted velocity from GMR. The velocity is scaled by 80 so that it remains within the kinematic limits of the robot. **V = [0, Vy/80, Vz/80]**. The scaling value is found using trial and error. It is chosen in such a way that it works for most letters.

  Inverse kinematics is performed using jacobian. The joint positions for each time step are calculated to control the robot.

  The maximum and minimum of the joint velocities are observed to be within [**-0.8373, 0.8373**] which is the velocity limit of all the joints in the robot [4].

  The algorithm is coded in such a way that it works only if the condition for velocity is satisfied.
- The trajectories are plotted and the robot is made to follow the trajectories of 3 letters.

4 Justification of approach

The video of the simulation is attached to the assignment. The simulation of 3 curved letters, O, U, and C, and 3 letters with straight lines I, L and M are displayed in the video. The velocity of the robot is scaled in such a way that it is well within the range. The scaling constant of 80 is chosen by considering the maximum and minimum velocity of different letters. The maximum and minimum velocity of a few letters are provided in the table below.

Letter	Max velocity	Min velocity
O	0.4964	-0.4713
U	0.3337	-0.3848
C	0.3297	-0.4086
I	0.2372	-0.0022
L	0.2746	-0.5480
M	0.5289	-0.5611

From the table you can observe that the chosen scaling constant works reasonably well for these letters as the maximum and minimum joint velocities are well within our range.

5 Limitations

- The algorithm does not work well for all the letters. The EM algorithm does not learn well for few letters like K. The prediction of Gaussian Mixture Regression is given in figure 2. From figure, we can observe that at some time steps, velocities are not present on the graph. The robot does not track the trajectory properly due to the inefficiency of the model.

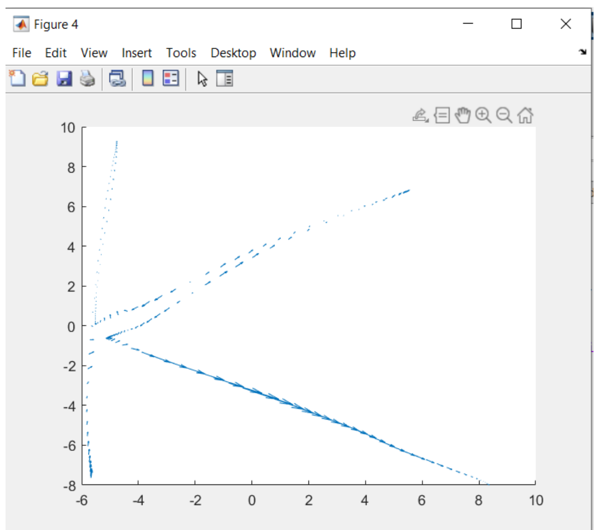

Figure 2: Output of Gaussian Mixture Regression of letter K

- The joint velocity limit of Kinova Gen3 is very low [-0.8337, 0.8337]. So, the maximum of the joint velocity of some letters can go beyond the limit.
- As explained in the lectures, the convergence may not be at the same spot as in letter G. Better learning models can be used for accurate predictions.

References

[1] InitGMM-Kmeans

[2] EM-GMM

[3] GMR

[4] Kinematic constraints of Kinova Gen3