Programming Assignment 1 – Part 2: Dynamics Movement Primitives (DMPs)

Due: Wednesday, February 24th 11:59pm

In the previous part of this assignment, you have developed a graphical user interface for capturing and pre-processing human demonstrations. The objective of the second part of this assignment is to study the behavior of one LfD representation, Dynamics Movement Primitives (DMPs). You have been given an implementation of DMPs that includes the following files:

a. `dmp.py` includes the implementation of the DMPs as a python class called `DiscreteDMP` that includes functions for integration, derivation, the canonical system, and the transformation equations (both the original [2002] and the improved [2009] set of equations). All the hyper-parameters also can be set in this file. This class uses Local Weighted Regression (LWR) from the `lwr.py` for solving the optimization/regression problem.
b. `lwr.py` includes the implementation of the Local Weighted Regression as a python class called `LWR` that includes basis function definition, learning of the parameters, and prediction.
c. `plot_tools.py` includes a function for plotting a given trajectory (including position, velocity, and acceleration).
d. `learn_batch.py` includes script for initializing, learning, reproducing, and plotting of a given demonstration. You can run this function to see an example of a learning, reproduction, and adaptation of a given demonstration.
e. `traj_full.json` is a json file including one captured demonstration. The file includes position data (x, y, z) together with frequency of recording in each axis.

You should first take a close look at the code and figure out how different phases of the algorithm including initialization, learning, reproduction, and adaptation work. Then perform the following steps:

a. Adapt the code to work with your h5 file formatting (instead of json). In other words, the given DMP implementation should be able to open and read your recorded demonstrations.
b. Collect 5 demonstrations of 10 different skills, pre-process and store them in 10 separate h5 files. You should write one function that allows you to load a single demonstration from a selected skill. One way is to save each skill (including 5 demonstrations) using the term skill (e.g. skill_02.h5). Make sure your preprocessing phase includes resampling and smoothing.
c. Try the implementation with one random demonstration and reproduce the skill from both the original initial point and a new initial point. First, keep the new initial point close to the original initial point. Then try to test with new initial points that are farther from the original initial point. For each reproduction, tune the DMP parameters and do not rely on the original parameters within the files.
d. Repeat the previous step for the goal point (final point). Then assign one new initial and final point for each demonstration in each skill.
e. Write a script that runs through all the sets and learns a DMP for each demonstration in each skill. The output should be a set of 10 plots each on including 5 demonstrations of the skill together with their reproductions once from the original (start or goal) point and then from the

new (start or goal) point. Make sure the plotted trajectories in each plot are consistent with other plots clearly distinguishable. Your plots should have title, xlabel, ylabel, and legends.

f. Define a new cost function and optimize the tuning phase. The cost can be built based on a dissimilarity metric. More discussions about this step in class.

PYTHON PACKAGES

You are allowed to use `numpy`, `scipy`, `h5py`, and `pygame` or `TkInter` for this assignment. Use of any other library should be discussed first. The given DMP implementation uses `Pylab` for plotting. You can, however, use `Matplotlib`.

Submission: By 11:59pm on Wednesday February 24th 2021, submit **only** one zip file through **Blackboard**. Make sure you organize the files in separate folders. For example, you can keep the data in a separate folder and move the class definitions into another folder while keeping the main file in the main folder. Your submission should include a PDF that includes the 10 plots described above. Since the plots include a lot of information, it is a good idea to keep one plot per page.