Learning for Adaptive and Reactive Robot Control
Solutions for Practical 3 - Simulation

Professor: Aude Billard
Contact: aude.billard@epfl.ch

Part 1: Experience compliant robot behavior in simulation

TASK 1 - Interact with the simulator

3. - Yes.
   - Yes.
   - No.
   - It should keep going until it reaches a joint or physical limit.

TASK 2 - Interact with the controlled robot

3. - They remain in the same configuration thanks to the null-space controller.
   - The robot goes back to the default configuration due to the null-space controller.
   - All joints are controlled by the null-space controller with a small gain. Only the Cartesian orientation of the end effector is actively controlled in IDLE mode. Its position is uncontrolled and can, therefore, be freely changed by the user.

TASK 3 - Program a linear DS in task space

10. - Yes, this method covers the entire workspace. It can handle any perturbations as it is globally convergent.
    - The DS will generate a straight-line trajectory for the end effector to pass through the base of the robot, which the robot cannot execute due to physical constraints. There are many solutions to remedy this; one of the simplest is to define the base of the robot as an obstacle so the DS creates trajectories to avoid it.
    - No. Due to uncertainties in the torque controller, the robot often deviates from the original trajectory. However, because the DS is globally convergent, the robot will still reach the target with a similar trajectory.
    - The error can be computed as the Euclidean distance between the target position and the end-effector position. It should be of the order of a few millimeters.
    - Yes. Compliance arises from torque control and global convergence from the DS planner.

Part 2: Learning from Kinesthetic Demonstrations

TASK 1 - Learn a DS with your own demonstrated trajectories

1. - This is case-dependent. SEDS should fit your demonstration relatively.

- No, you need to provide demonstrations spanning the entire workspace for the algorithm to generalize to the entire workspace.

4. - Yes.
   - Yes, it should. In some rare cases, the robot can become stuck before reaching the target due to uncertainties in the controller moving it to spots in the workspace with low velocities, due to not having demonstrations there. When converted to torques, the low velocities might not be enough to move the robot out of the spot.

6. - Yes, it should. The main challenge in kinesthetic demonstrations is usually the speed of the end-effector when recording. Because the dataset used to learn a DS uses both the position and velocity of the recorded trajectories, speed is crucial when recording. Accelerations and full-stops will hinder your learned DS and create areas with very high or very low speeds. This won't affect the trajectory generation, however the robot might be unable to exactly match the desired changes in speed, especially when controlling it in torques to ensure compliance. This often leads the robot to move outside of the demonstrated workspace, resulting in not following the demonstrations.
   - You need enough trajectories to span the workspace. Intersecting trajectories will cause issues in the learned DS as a single point cannot have multiple directions. The velocity of the robot when recording also plays a crucial role. In short, it is important to have high quality trajectories spanning the workspace, without intersecting, to generalize well.
   - A DS learned with low-quality recordings will often suffer with velocity variations which the controller cannot track. This is due to the DS learning with the velocity of the recordings. To ensure replicable trajectories, one must demonstrate smooth trajectories both in position and velocity. Not generalizing results in unexpected behavior in the untaught ares of the workspace.

7. - Yes.
   - Yes.
   - Yes, it should.

TASK 2 - Add an obstacle to your environment and modulate your DS to avoid it

6. - No.
   - No.

7. - No, it now avoids the obstacle.
   - Yes.
   - Depending on the DS learned, it should follow a new path more or less similar to the learned one. Unless, stuck due to a combination of low DS velocities and controller uncertainties, the robot should always reach the target.

8. - No, it should now pass closer to the obstacle.
   - Yes.

Part 3: Obstacle Avoidance

TASK 1 - Avoidance of a single obstacle with a linear DS

3. - No.
   - No.
4. - It should now avoid the obstacle.
   - Yes.
   - Yes.
5. Increasing the sensitivity value increases the weight of the modulation to avoid the obstacle. This makes the robot move further away from the obstacle to avoid it.
6. No. This will make the attractor impossible to reach as the robot cannot pass through the obstacle. However, due to discretization, uncertainties in the robot position and the surface of the obstacle begin very thin, the end-effector can sometimes end up inside the obstacle, and thus is able to reach the target.
7. - The robot should react to the new position and avoid the obstacle.
   - Yes.
   - The depends on the frequency of the control loop, which can be slowed down due to hardware specs. It is technically possible to move the obstacle moves so fast that the control loop does not have time to register its new position and avoid it. This is due to hardware and communication between MATLAB, the control loop and the simulation.

TASK 2 - Avoidance of multiple obstacles

1. No. Only the end effector is avoiding the obstacle. This is due to using a 3D Cartesian DS to plan the trajectory. This DS controls a single point in Cartesian space, representing the end effector of the robot. Because the trajectory planner (DS) is not aware of the rest of the robot, it cannot make it avoid an obstacle.
2. This depends on how you placed the obstacles.
3. This depends on the obstacles. You can do it with a single one if its big enough.

TASK 3 - Avoidance of complex obstacles

3. Depending on the obstacles, it should. Because the robot is now controlled in joint space, the entire robot arm will move to avoid the obstacles.
4. Yes.
6. The robot should avoid the line, wall and ring obstacles correctly. It will be stuck on the tshape obstacle
7. The tshape obstacle is concave, meaning that to avoid it, the robot must move in a direction opposite to the nominal command. These types of obstacle therefore cannot be avoided with a simple DS modulation. You should however be able to guide the robot out of the concave obstacle.
9. Yes.
10. This depends on the new obstacles.