Self Collision Avoidance Performance

This directory contains the simulation comparisons for Self-Colision enabled and disabled scenarios.

Two simulations are included, to demonstrate the performance of self collision avoidance.

Simulation 1 : Inputs

initial configuration = **[math.pi/2, 0.0, 0.0, 0.0, -0.35, -0.698, -0.505, 0.0, 0.0, 0.0, 0.0, 0.0]**
Tsc initial = **[0.7, 0.7, math.pi/4]**
Tsc goal = **[0, 0, math.pi]**
gains = **[0.18, 0.00015**]

Simulation 1 : Cube Configuration

	X (meters)	Y (meters)	Phi (radian)
Initial	0.7	0.7	0.78539
Goal	0	0	3.14159

Simulation 2 : Inputs

initial configuration = **[0.0, 0.0, 0.0, 0.0, -0.35, -0.698, -0.505, 0.0, 0.0, 0.0, 0.0, 0.0]**
Tsc initial = **[1, 0, 0]**
Tsc goal = **[0, 0, 0]**
gains = **[0.18, 0.00015**]

Simulation 2 : Cube Configuration

	X (meters)	Y (meters)	Phi (radian)
Initial	1	0	0
Goal	0	0	0

Self Collision Avoidance

Joint value limiting was implemented at first, according to the Capstone guide, at *http://hades.mech.northwestern.edu/index.php/Mobile_Manipulation_Capstone*

However, as the guide suggests, "Your approximation should be conservative, meaning that allowed configurations are never in self-collision. But it should not be so conservative that the arm's workspace is overly constrained, preventing the robot from doing useful work”. After testing with CoppeliaSim Scene3_youBot, I realized that it is difficult to find joint limits that guarantee no-self-collisions without restricting the robot’s feasible task space significantly.

Hence I have developed another way to avoid self-collisions, without imposing strict joint limits.

1. The robot base and the arm base is modeled by a cuboid and a vertical cylinder.
2. This modeled space is designated as restricted space for the end effector. By observing youBot, we can guarantee that if the end effector is not within the restricted space, there cannot be any self collision.
3. At each ‘nextConfiguration’ simulation using the youBot kinematic simulator, I evaluate whether the next configuration penetrates the restricted space.
4. If nextConfiguration penetrates the restricted space, the robot Jacobian columns relevant to robot arm (5 columns) are set to zero and the controls are calculated again. When the arm jacobian columns are set to zero, the robot controller will not request any motion from the robot arm, instead it will try to generate the required twist by changing the chassis configuration.

Controller

kp = 0.18
ki = 0.00015

The implemented controller is a feedforward + feedback PI controller

Results files

youBot_Trajectory.csv

youBot_Trajectory file will contain the configuration of the youBot end effector, at each 0.01 second time interval. Comments will be printed in the file, indicating the specific format of the configuration (phi, x, y, J1, J2, J3, J4, J5, W1, W2, W3, W4). ***The youBot_Trajectory.csv file can be loaded into CoppeliaSim Scene6_youBot_cube.***

trajectory_Xerr.csv

trajectory_Xerr file will contains the error-twist, at each 0.01 second time interval. Comments will be printed in the file, indicating the specific format of the error (Wx, Wy, Wz, Vx, Vy, Vz)