In [209…

```python
import modern_robotics as mr
import numpy as np
```

In [ ]:

```python
def TrajectoryGenerator(T_se_i, T_sc_i, T_sc_f, T_ce_grasp, T_ce_standoff, k: int):
    # trajectory runtime (s)
    Tf = 1

    # number of configurations
    N = int( (Tf * k) / 0.001 )
    output = np.zeros((6 * N + 2, 13))
    idx = 0

    # Trajectory to standoff position
    T_se_standoff_pick = np.matmul(T_sc_i, T_ce_standoff)
    traj = mr.ScrewTrajectory(T_se_i, T_se_standoff_pick, Tf, N, 5)
    for t in traj:
        output[idx] = np.concatenate((t[:3, :3].flatten(), t[:3, 3], [0]))
        idx += 1

    # Trajectory to grasp position
    T_se_grasp_pick = np.matmul(T_sc_i, T_ce_grasp)
    traj = mr.ScrewTrajectory(T_se_standoff_pick, T_se_grasp_pick, Tf, N, 5)
    for t in traj:
        output[idx] = np.concatenate((t[:3, :3].flatten(), t[:3, 3], [0]))
        idx += 1

    # open the gripper
    output[idx] = np.concatenate((T_se_grasp_pick[:3, :3].flatten(), T_se_grasp_pick[:3, 3], [1]))
    idx += 1

    # move back to the standoff point for the block pick up
    traj = mr.ScrewTrajectory(T_se_grasp_pick, T_se_standoff_pick, Tf, N, 5)
    for t in traj:
        output[idx] = np.concatenate((t[:3, :3].flatten(), t[:3, 3], [1]))
        idx += 1

    # move to the standoff point for the block drop off
    T_se_standoff_drop = np.matmul(T_sc_f, T_ce_standoff)
    traj = mr.ScrewTrajectory(T_se_standoff_pick, T_se_standoff_drop, Tf, N, 5)
    for t in traj:
        output[idx] = np.concatenate((t[:3, :3].flatten(), t[:3, 3], [1]))
        idx += 1

    # place down the block
    T_se_grasp_drop = np.matmul(T_sc_f, T_ce_grasp)
    traj = mr.ScrewTrajectory(T_se_standoff_drop, T_se_grasp_drop, Tf, N, 5)
    for t in traj:
        output[idx] = np.concatenate((t[:3, :3].flatten(), t[:3, 3], [1]))
        idx += 1

    # open the gripper
    output[idx] = np.concatenate((T_se_grasp_drop[:3, :3].flatten(), T_se_grasp_drop[:3, 3], [0]))
    idx += 1

    # move the the standoff point above the drop location
    traj = mr.ScrewTrajectory(T_se_grasp_drop, T_se_standoff_drop, Tf, N, 5)
    for t in traj:
        output[idx] = np.concatenate((t[:3, :3].flatten(), t[:3, 3], [0]))
        idx += 1

    return output
```

In [ ]:

```python
import csv

# initialize the transformation matrices
T_se = np.array([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0.5], [0, 0, 0, 1]])
T_sc_i = np.array([[1, 0, 0, 1], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]])
T_sc_f = np.array([[np.cos(-np.pi/2), -np.sin(-np.pi/2), 0, 0], [np.sin(-np.pi/2), np.cos(-np.pi/2), 0, -1], [0, 0, 1, 0], [0, 0,
T_ce_grasp = np.array([[-1, 0, 0, 0], [0, 1, 0, 0], [0, 0, -1, 0.005], [0, 0, 0, 1]])
T_ce_standoff = np.array([[-1, 0, 0, 0], [0, 1, 0, 0], [0, 0, -1, 0.1], [0, 0, 0, 1]])

# create the csv file
with open("test.csv", "w+") as f:
    writer = csv.writer(f)

    traj = TrajectoryGenerator(T_se, T_sc_i, T_sc_f, T_ce_grasp, T_ce_standoff, 1)
    writer.writerows(traj)
```

Video Link

link