**Step-By-Step Assignment Instructionsless**

The UR5 is a popular 6-DOF industrial robot arm. The robot has geared motors at each joint, but in this project, we ignore the effects of gearing, such as friction and the increased apparent inertia of the rotor.

The relevant kinematic and inertial parameters of the UR5 are:

A close up of text on a white background

Description automatically generated

For your convenience, these parameters are expressed in Python, Mathematica, and MATLAB below.

[UR5\_parameter.py](https://d18ky98rnyall9.cloudfront.net/_be39b6c4a1f5e8e9109587e3e5a1ae56_UR5_parameter.py?Expires=1584835200&Signature=dz~ZCZnetIW98vMKspxgsZsuRxcKONuIV1oFRWjtkFqPXJKzw3jby8MXrDle3K-um5WZJ4s5Mb7Gizd0nrLeMZZT32L587cc1aGa9DqCNiMoQaLfhA~IVNtqF3gaPFHKwZs9gGp3cKftZFjS2dRFE37BgzQJ6KwfASRUJig8Oa8_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

[UR5\_parameters.nb](https://d18ky98rnyall9.cloudfront.net/_df2bcf0f658e6f852b0d42174ffce152_UR5_parameters.nb?Expires=1584835200&Signature=hpHIRbcj2ufWk6xIn0c~jksgucf5ATde-W9emkBkTm56EPsubwfrwLNPOuHwerpwZmk20IaWHAFM1N4RuoPnLr4aH5zwehJHI9NEURz6rp8PUtozrsdpPG4XoaKyr2uLbi8KvDyWJfiRpjI8j~EVER1Ab8Gf~~CJMpXocaoYqU0_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

[UR5\_parameters.m](https://d18ky98rnyall9.cloudfront.net/_fcfea4995ab55579739fd5e3e1510e43_UR5_parameters.m?Expires=1584835200&Signature=Lp4hq5a42LNPLCryhME7VepEqoz3f2et38Mr~U~~1F1PM3MZvOoVQU8wb4pbHRjYTAqy1S0XqERLCxlIJN7YlmOsufTIX0h3S3ovX0tJijV~Ed03In5KWxv1WpPJpH4TEHt6d02JbB2Ua3~JZVQ98kAQW0rOKAx0N5LjC5sXfY4_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

Your job is to write code that simulates the motion of the UR5 for a specified amount of time (in seconds), from a specified initial configuration (at zero velocity), when zero torques are applied to the joints. In other words, the robot simply falls in gravity. Gravity is g = 9.81*g*=9.81m/s^22 in the −z^-direction, i.e., gravity acts downward. The motion should be simulated with at least 100 integration steps per second. Your program should calculate and record the robot joint angles at each step. This data should be saved as a .csv file, where each row has six numbers separated by commas. This .csv file is suitable for animation with the CoppeliaSim UR5 csv animation scene.

You will perform two simulations and make videos of each:

1. The robot falling from the zero (home) configuration for 3 seconds.
2. The robot falling from a configuration where all joints are at their zero position, except for joint 2, which is at −1 radian. This simulation should last 5 seconds.

**Important:** Since the simulated robot has no friction and zero motor torques, no energy is added or subtracted during the simulated motion. Therefore, the total energy of the robot (kinetic plus potential) must remain constant during the simulation. If you see the robot swinging higher and higher, or losing energy, something is wrong wih your simulation.

**Your submission should be a single .zip file of a directory with the following contents:**

1. **Your commented code in a directory called "code."** Your code should be lightly commented, so it is clear to the reader what the code is doing. No need to go overboard, but keep in mind your reviewer may not be fluent in your programming language. Your code comments must include an example of how to use the code. Only turn in functions that you wrote or modified; you don't need to turn in other MR functions that your code uses. If your code is in MATLAB or Python, just turn in the text files with your functions. If your code is in Mathematica, turn in (a) your .nb notebook file and (b) a .pdf printout of your code, so a reviewer can read your code without having to have the Mathematica software. **Your code must automatically create the .csv file of the simulation.**No cutting and pasting output printed to the screen to create your .csv file.
2. **A file called simulation1.csv.**This is the .csv file created by your code for the first scenario.
3. **A CoppeliaSim animation called simulation1.xxx (where xxx should be a common video format, like mp4).** Use the CoppeliaSim csv animation scene for the UR5 (Scene 2). The video should show CoppeliaSim "playing" your .csv file. (See the CoppeliaSim description in module 1 of Course 1, or go directly to <http://hades.mech.northwestern.edu/index.php/CoppeliaSim_Introduction>, to learn about making videos with CoppeliaSim.) Your video should be a "reasonable" size (e.g., a few MB, less than 10 MB) and use a standard codec that others can view. Your video should be taken from a virtual camera angle that makes it easy to see the end-effector configuration during the motion. Preferably, you should choose the time multiplier in the CoppeliaSim playback so that the motion is approximately real time (i.e., the scenario 1 animation is approximately 3 seconds and the scenario 2 animation is approximately 5 seconds).
4. **A file called simulation2.csv.**This is the .csv file created by your code for the second scenario.
5. **A CoppeliaSim animation called simulation2.xxx (where xxx should be a common video format, like mp4).**
6. **(OPTIONAL) A plain text file called "README.txt" or similar.**This has any other information that may help the reviewer understand your submission.