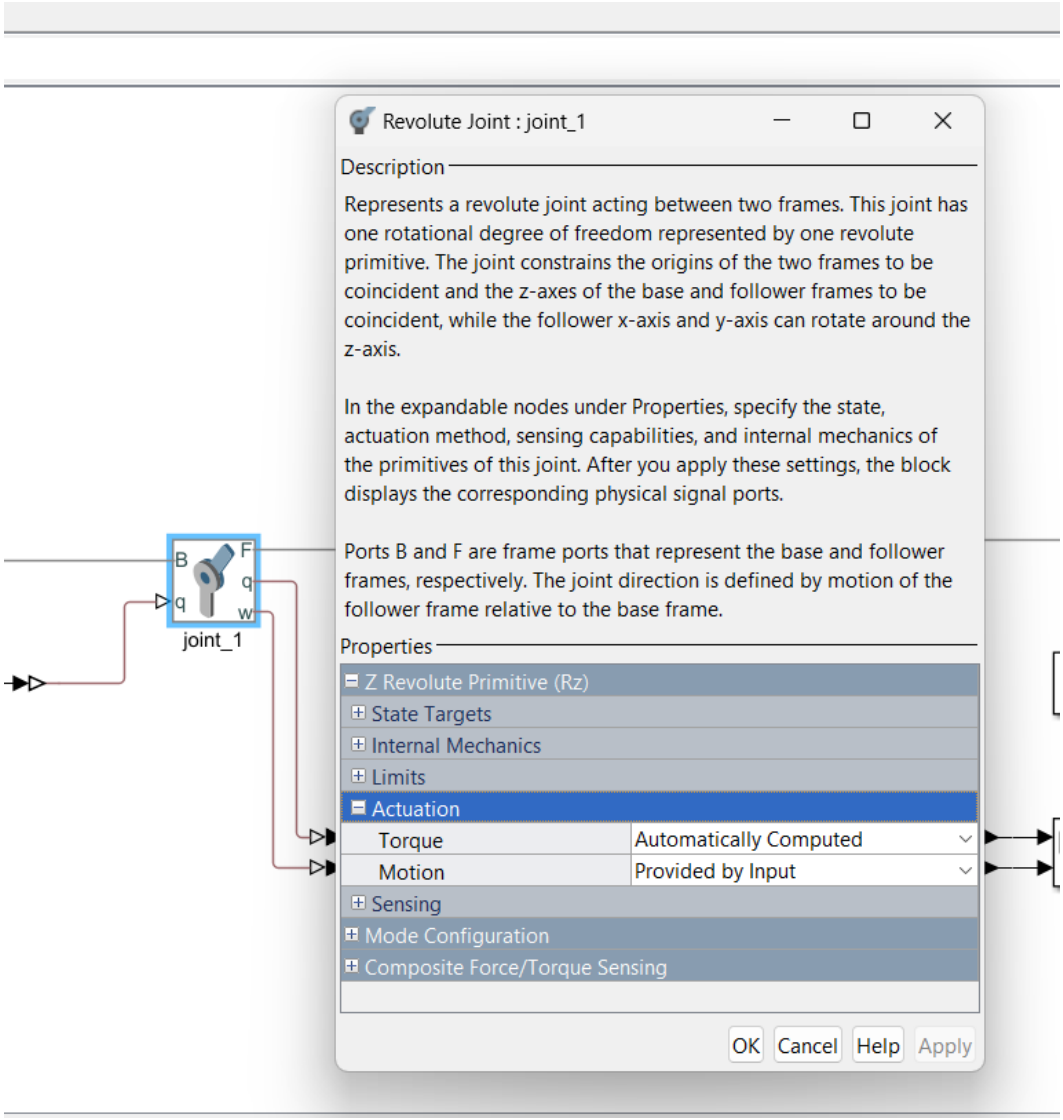
When creating a URDF file, it's important to structure the assembly files so that non-moving parts within the assembly are represented as single part files.

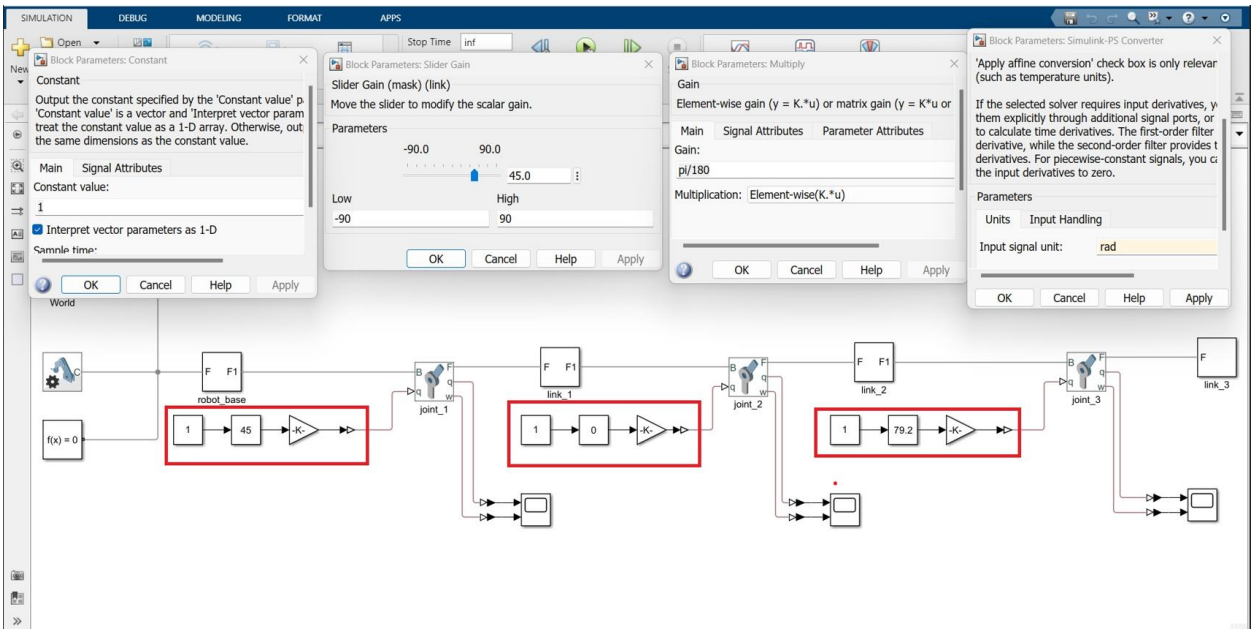
The URDF files are imported in Simulink using smimport(“filename.urdf”) in the command window in MATLAB.

To simulate the model using slider gain

Open each joint block and set parameters as given below.

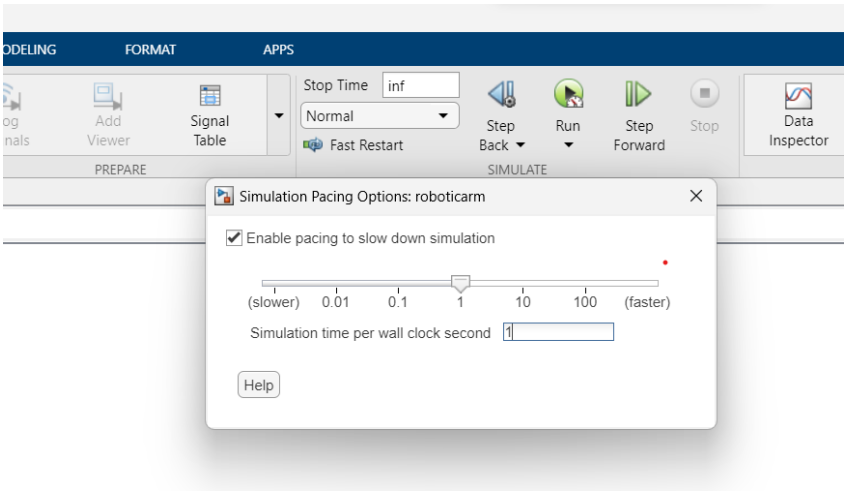


To actuate joints, add constant, slider and multiply block with Simulink to PS converter, which is joined to motion input block as shown below: -

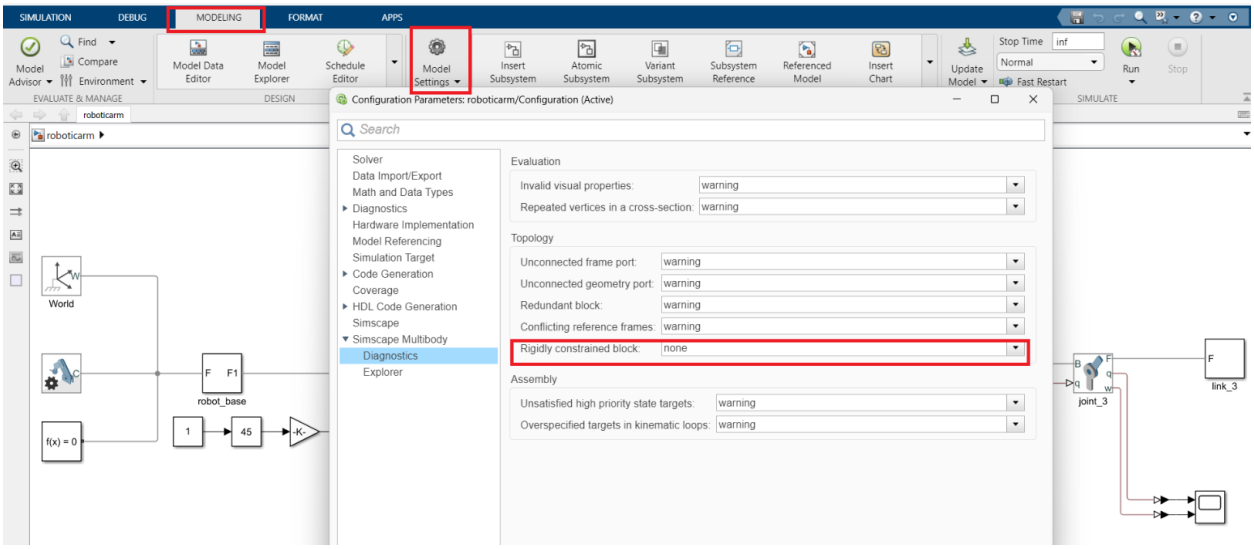


Then, if you want to see the position, velocity and other parameters, you can enable sensing and add PS to the Simulink converter with scope to visualize them better.

Now, to run the simulation you have to change the settings as given below, i.e. set Stop Time to inf and enable pacing to slow down the simulation.



Also, in Modeling---->Model Settings---->Diagnostics
Change the rigidly constrained block to none from error.

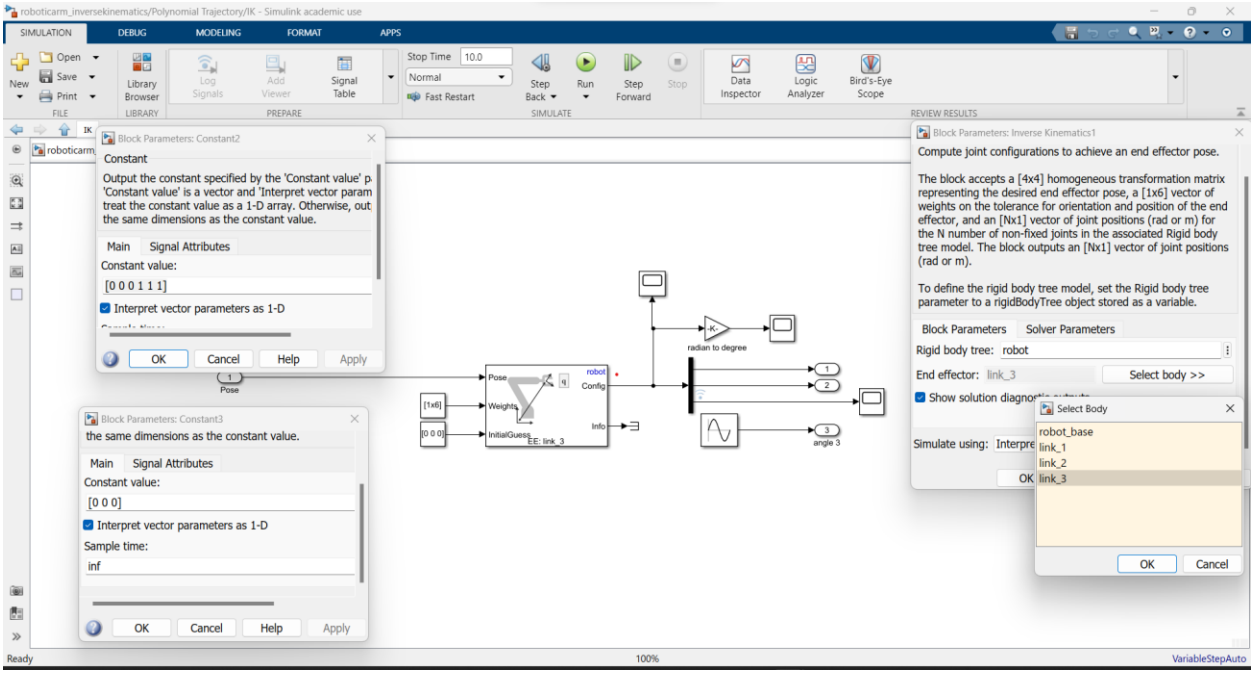


Now, you can run the simulation and see the robot moving when you change the slider.

Now, to simulate the robot through inverse kinematics and trajectory planning:

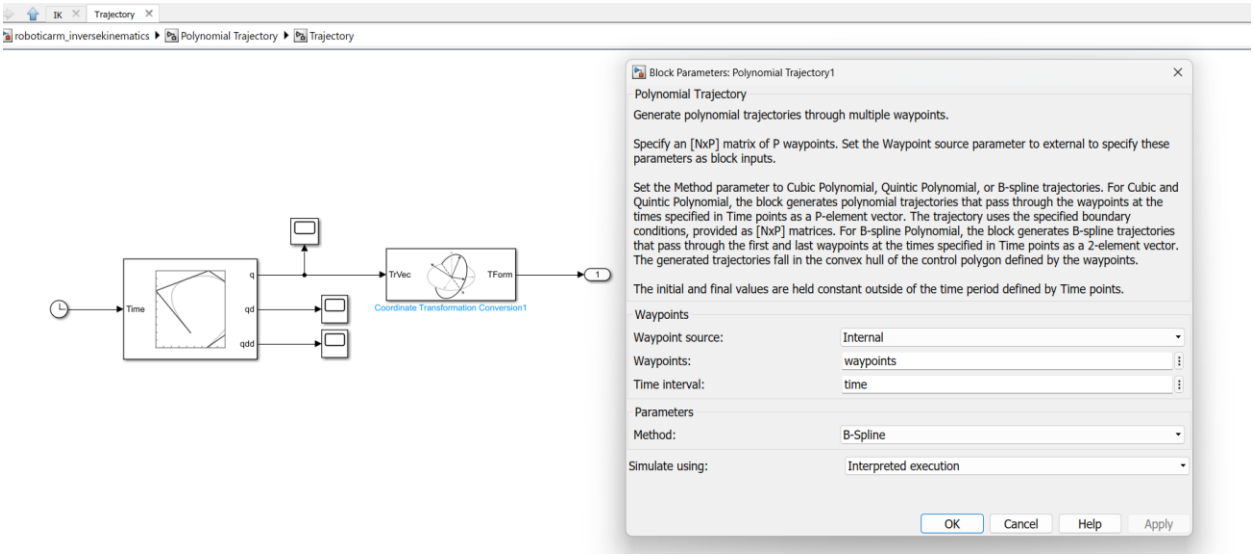
Do the same as done previously to import your model in Simulink using smimport, then, in the MATLAB command window, run the command
robot = importrobot("filename.urdf")
This will load your URDF file in the MATLAB workspace.

Then open your Simulink model and add inverse kinematics block in it and set parameters as given below:



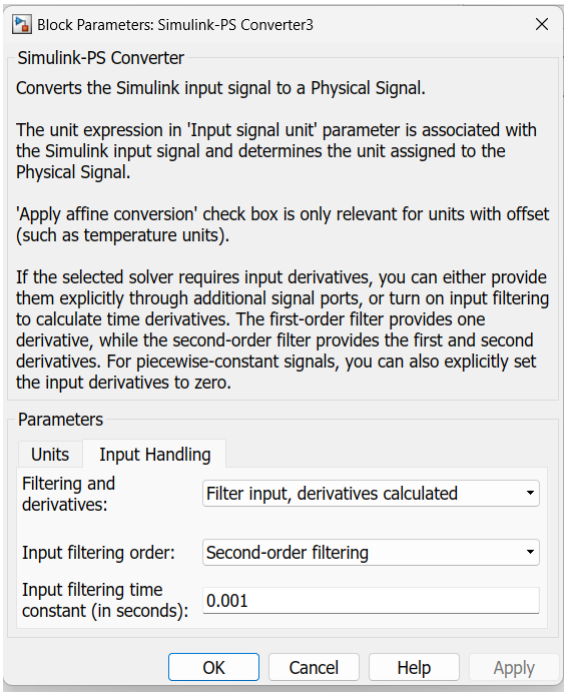
Add a terminator block to info and a demux to config block to give configuration to our joints.

Now, add a polynomial trajectory block to the model with a coordinate transformation block. Set the parameters as given below:

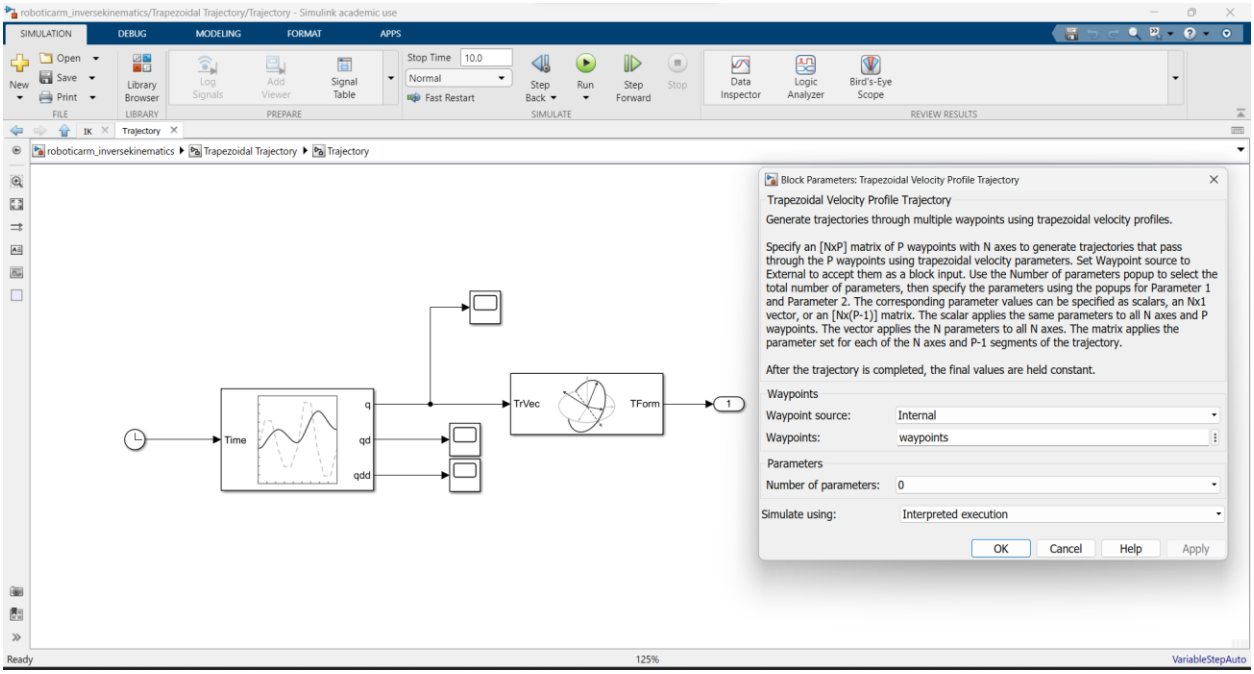


Now, in the MATLAB command window, add waypoints and time in the workspace
Note that waypoints must be in the NxP matrix where N is row and P is waypoints, i.e., N contains the X, Y, Z axes in rows, and P contains number of waypoints in columns.

Now connect the output of the coordinate transformation block to the pose of the inverse kinematics block and the output of the inverse kinematics block, i.e. configs via demux to all the joint angles.
Note: the input handling of the Simulink to PS converter must be set as given below



Similarly, we can use a Trapezoidal velocity profile trajectory just in place of a polynomial trajectory as given below:



You can use scope to visualize data of various blocks and results.