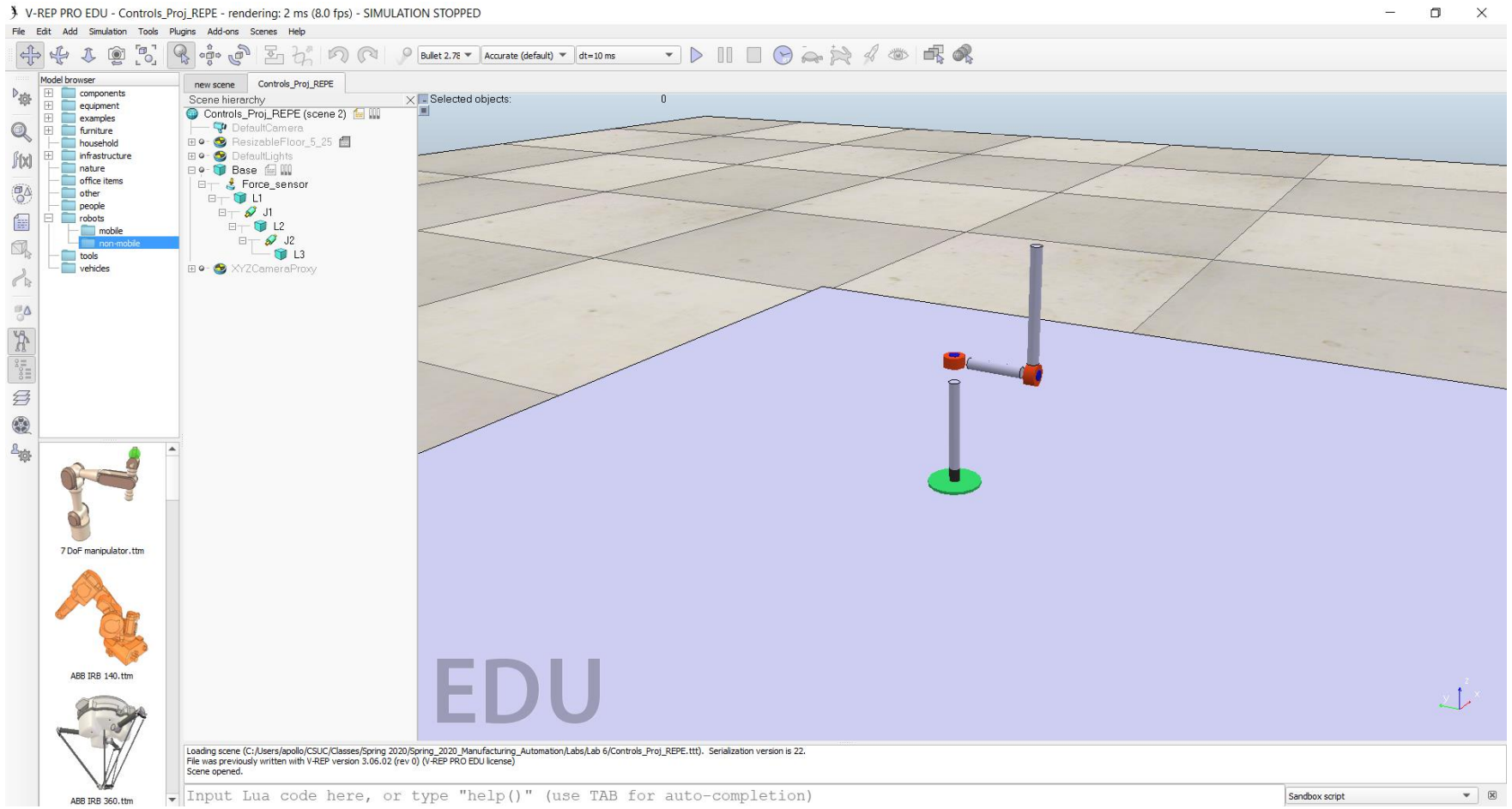
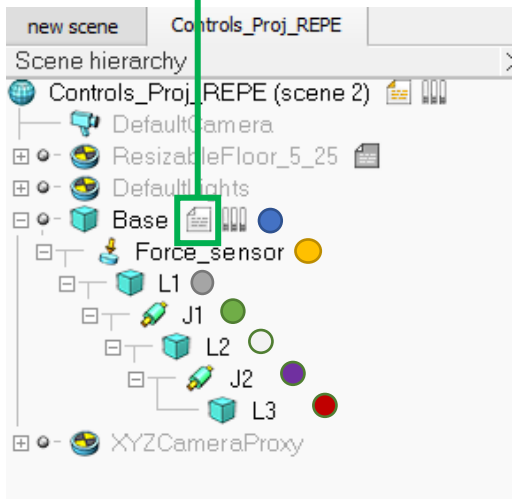
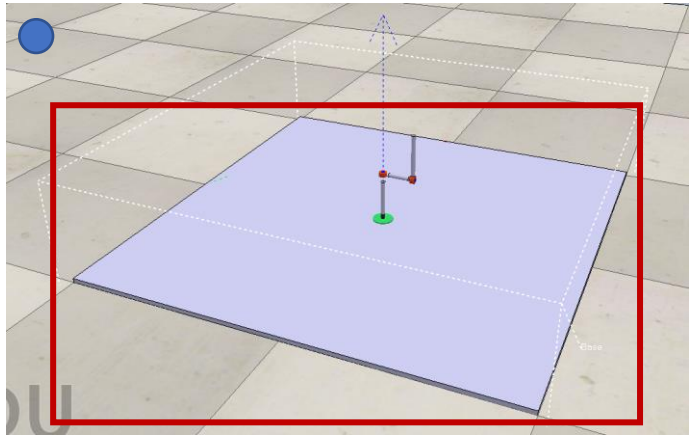
?? How to connect MATLAB/ Simulink and Coppelia ??




Furuta Pendulum CoppeliaSim (former VREP) Model

Hierarchy for the Model



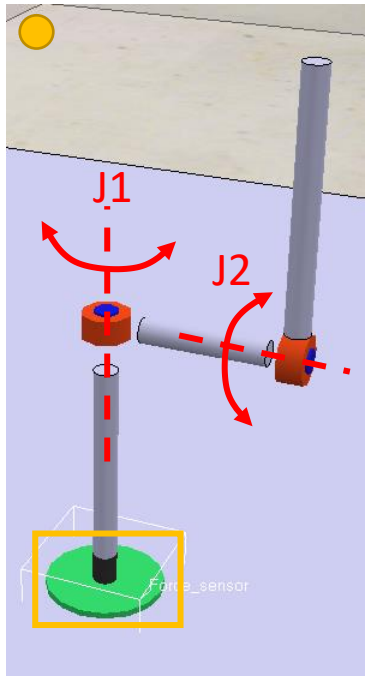



Base

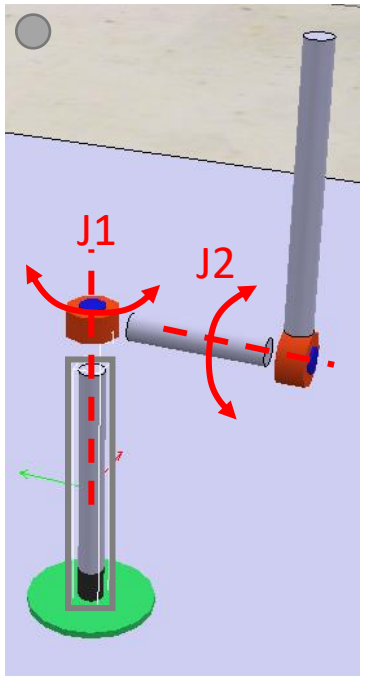
Non-threaded Child Script
to connect MATLAB

```
function sysCall_init()
    simRemoteApi.start(19999)
end
```

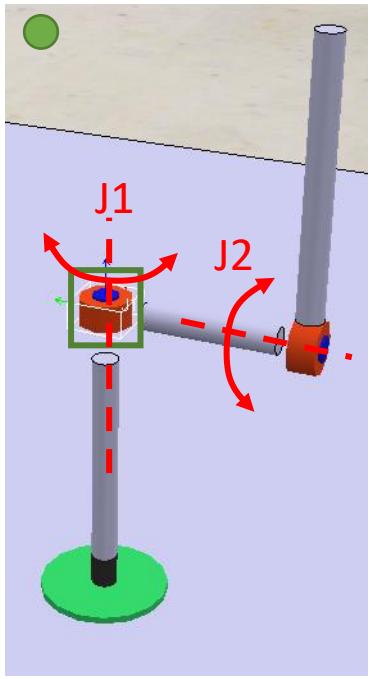
In CoppeliaSim



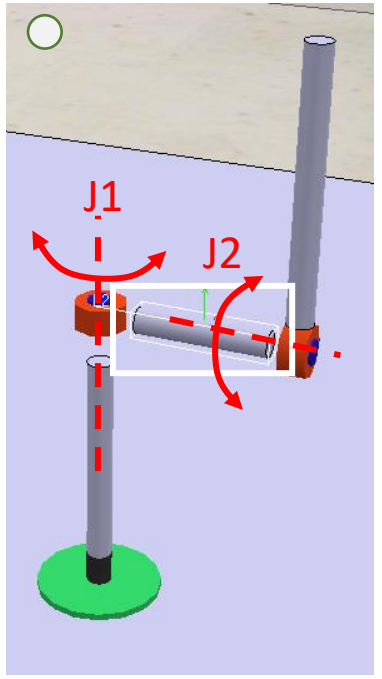

Force_sensor



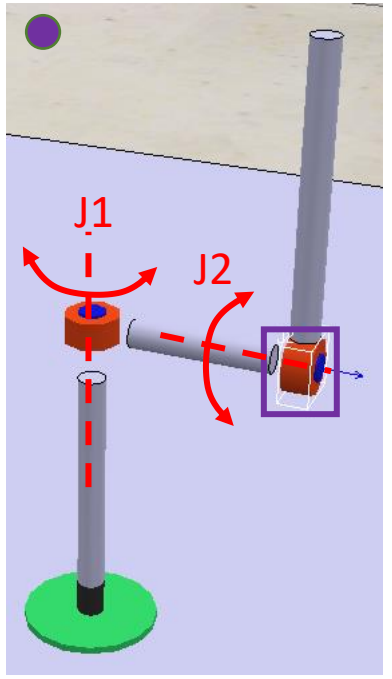

L1



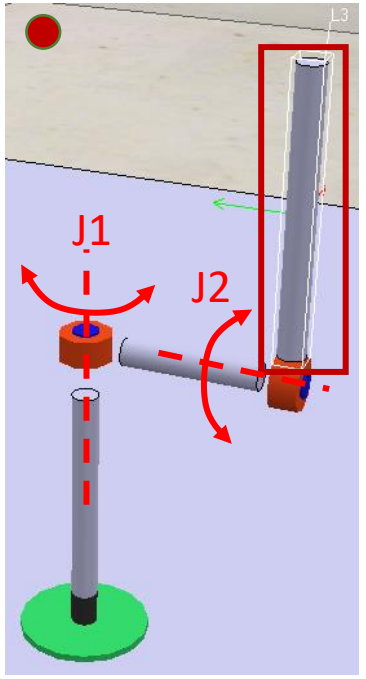

J1




L2




J2




L3

Make sure you have following files in your directory, in order to run the various examples:
({Coppelia Installation Folder}/remoteApiBindings/matlab/matlab/)

1. remApi.m
2. remoteApiProto.m
3. the appropriate remote API library: "remoteApi.dll" (Windows), "remoteApi.dylib" (Mac) or "remoteApi.so" (Linux)
4. simpleTest.m (or any other example program)

    Add the connection IP and port. IP could be localhost (127.0.0.1) –or your network IP- and PORT must be the same as in the port that you specified in the non-threaded child script (e.g., in earlier slide it was 19999).

---

If you choose not to use the prototype file ("remoteApiProto.m"), then you will have to make sure you have a compiler set-up for Matlab (mex -setup). You will also need "extApi.h" in this folder, and you will have to instanciate the remote API with "sim=remApi('remoteApi','extApi.h');" instead of "sim=remApi('remoteApi');"

Finally, if you wish to rebuild the prototype file, you will have to comply with above conditions, then type:

loadlibrary('remoteApi','extApi.h','mfilename','remoteApiProto')

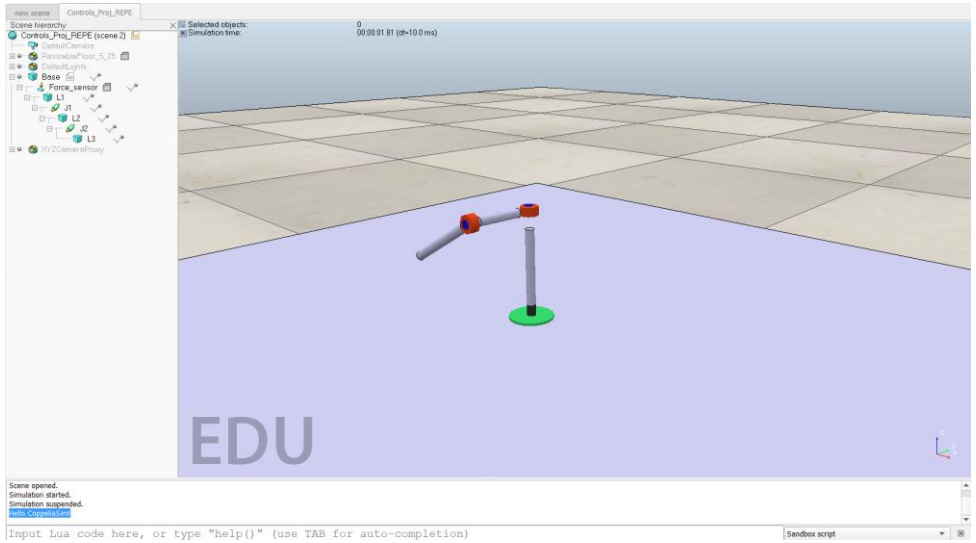
For more examples, have a look at the python folder: language is different but principles are the same

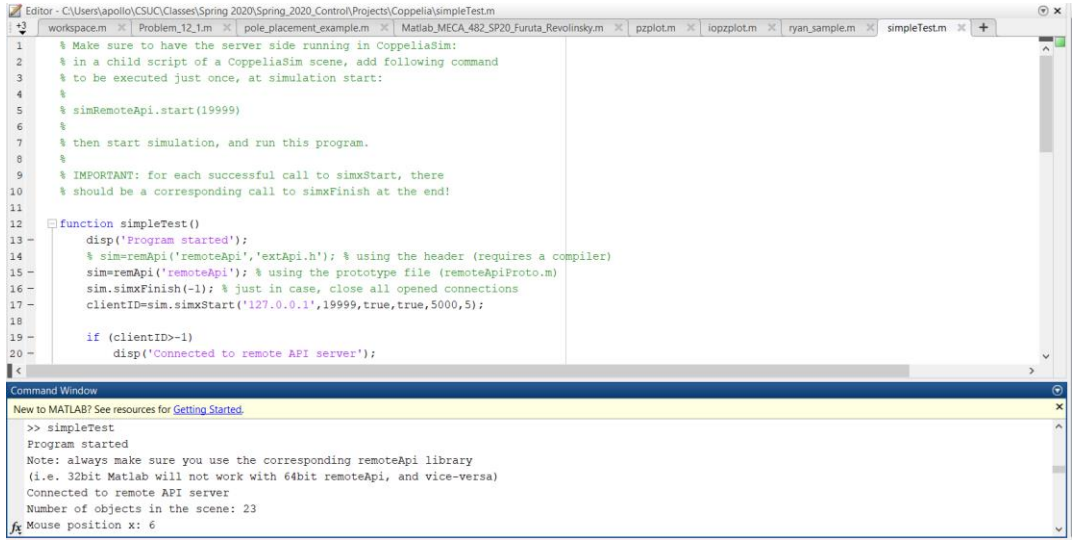
# Project Folder

| Name | Date modified | Type | Size |
|---|---|---|---|
| remApi.m | 1/10/2020 6:30 AM | M File | 100 KB |
| remoteApi.dll | 1/10/2020 6:30 AM | Application extens... | 76 KB |
| remoteApiProto.m | 1/10/2020 6:30 AM | M File | 36 KB |
| simpleTest.m | 11/12/2019 4:25 AM | M File | 3 KB |

- {Coppelia Installation Folder}/programming/remoteApiBindings/lib
- {Coppelia Installation Folder}/programming/remoteApiBindings/matlab/matlab

To test the connectivity one can use the test files (e.g., simpleTest.m) and open it in MATLAB.

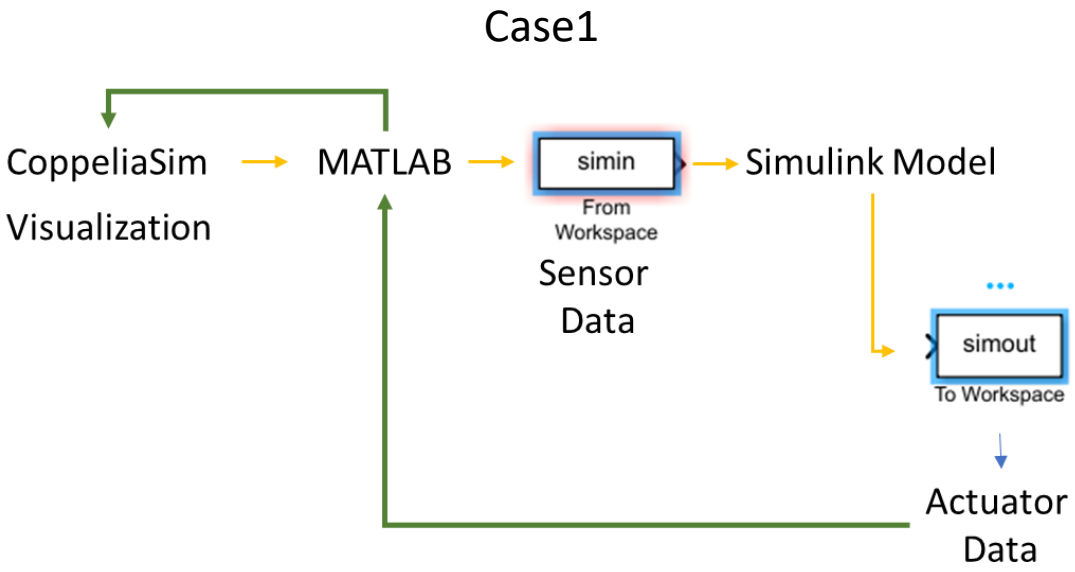


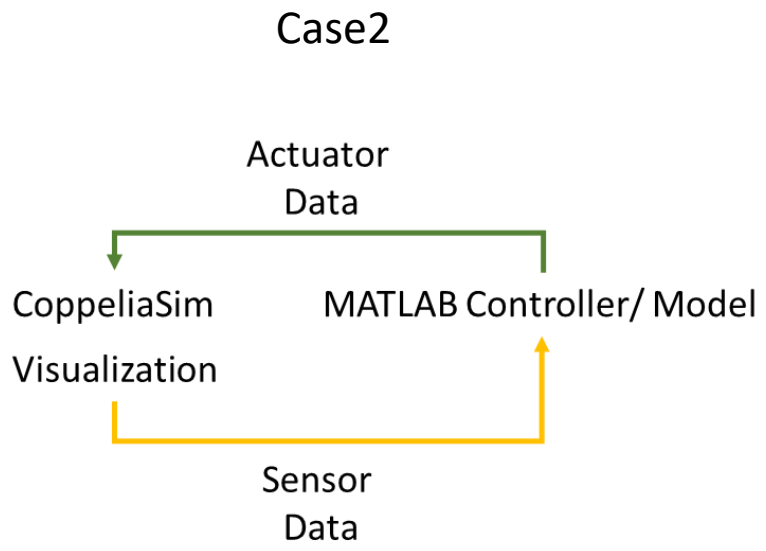



Collect sensor data (e.g., position, velocity, etc.) from the Coppelia and store it in a variable.

Case1: If your model is in Simulink add a block to communicate with MATLAB continuously

Case 2: If your model and controller is in MATLAB use directly stored data and send the actuation signal back to Coppelia.



Collect sensor data (e.g., position, velocity, etc.) from the Coppelia and store it in a variable. The syntax below is from Python and can be used in MATLAB with minor changes.

```
#----- Step 1: Initialize Joint Handles where you defined your joint under the name of J1 ---------
err_code, j1 = sim.simxGetObjectHandle(clientID,  "J1"  ,sim.simx_opmode_blocking)

#----- Example setting a physical entity (the case below represent target velocity) ---------
vel_val = numpy.radians(  0  )   # Target velocity
#Set the position of J1
err_code = sim.simxSetJointTargetVelocity(clientID,j1,pos_val,sim.simx_opmode_streaming)
```

More details about API is in https://coppeliarobotics.com/helpFiles/en/apiFunctions.htm

**Q:** How are you going to balance the pendulum? When I started the simulation it doesn't give me the time to start MATLAB to trigger the controller.

**A:** Once you initialize the CoppeliaSim directly pause it and then trigger the CoppeliaSim from MATLAB script.