Useful links:

<http://www.v-rep.eu/helpFiles/en/remoteApiClientSide.htm>

<http://www.v-rep.eu/helpFiles/en/remoteApiServerSide.htm>

Need to have:

* Matlab 32 bit (64 will not work)
* V-rep
* C compiler for matlab

Launch Matlab. In matlab console, type ‘mex-setup’. This will help you set up the C-compiler. Copy the following files into your current directory (they are located within the vrep install directory /programming):

* extApi.h
* remoteApi.dll
* simpleTest.m

Run simpleTest.m to ensure that the compiler is working (it will not successfully connect, but will not give you errors). This is the client side set up. Now to set up the server side, open the file remoteApiConnections.txt from the vrep install directory. Add the lines:

portIndex1\_port = 19999

portIndex1\_debug = false

portIndex1\_syncSimTrigger = true

Now launch vrep. Return to matlab and run simpleTest.m again. This time it should connect and tell you how many objects are in the scene. Now the server and client side are set up. Enjoy!

Ubuntu bug:

add ‘.so’ to obj.libName to the loadlibrary() and unloadlibrary() call in remApi.m function

change path in cd() of task\_arm\_path\_solver to the directory for trajectories and cost variables

make soft links from vrep home to .ttt files in tasks/manipulation/vrep\_scenes

the script for reading the trajectory and writing the cost vars is within the ttt files

In the script turn on/off simulation with flag line 87

Based on the scene: forwardandinversekinematics2.ttt in the scenes file

remApi.m: In the function simxStart, add default for commThreadCycleInMs

in task\_arm\_path\_solver, simxLoadScene requires clientID as first argument

Almost everything works => arm\_goal\_example connects, makes vrep load file [buggy, better to have file already loaded], starting simulation, freezes v-rep ☹. Starting simulation within vrep does work.

if stopping Matlab before the end, then run task\_solver.close\_sim() to close down the connection to VREP