

Seminar Robot-assisted Surgery in Clinics

Surgical robot path planning & simulation based on V-rep

MATLAB – V-REP Communication

Daniel Gheorghita, MSc. Informatics Student

1. V-REP side connection

- select an object from the environment which will be present anytime (for this example, STL_HumanEye is used). Hint: try not to select auxiliary objects which might not be needed (hence deleted) at some point.
- add a child script to the selected object if it does not already have one (right click object → Add → Associated child script → Non threaded)
- insert in the initialization call the following line:

```

1  -- DO NOT WRITE CODE OUTSIDE OF THE if-then-end SECTIONS BELOW!! (unless the code is a function
2
3  if (sim_call_type==sim_childscriptcall_initialization) then
4      simExtRemoteApiStart(19999)
5      -- Put some initialization code here
6
7      -- Make sure you read the section on "Accessing general-type objects programmatically"

```

- save the script and the scene

2. MATLAB side connection

- include the API generated by V-REP, close all connections (if any, as matter of precaution) and connect to the V-REP environment

```

vrep=remApi('remoteApi'); % using the prototype file (remoteApiProto.m)
vrep.simxFinish(-1); % just in case, close all opened connections
clientID=vrep.simxStart('127.0.0.1', 19999, true, true, 5000, 5);

```

simxStart

Description	Starts a communication thread with the server (i.e. V-REP). A same client may start several communication threads (but only one communication thread for a given IP and port). This should be the very first remote API function called on the client side. Make sure to start an appropriate remote API server service on the server side, that will wait for a connection. See also simxFinish . This is a remote API helper function.
Matlab synopsis	[number clientID]=simxStart(string connectionAddress,number connectionPort,boolean waitUntilConnected,boolean doNotReconnectOnceDisconnected,number timeOutInMs,number commThreadCycleInMs)
Matlab parameters	connectionAddress : the ip address where the server is located (i.e. V-REP) connectionPort : the port number where to connect waitUntilConnected : if true, then the function blocks until connected (or timed out). doNotReconnectOnceDisconnected : if true, then the communication thread will not attempt a second connection if a connection was lost. timeOutInMs : if positive: the connection time-out in milliseconds for the first connection attempt. In that case, the time-out for blocking function calls is 5000 milliseconds. if negative: its positive value is the time-out for blocking function calls. In that case, the connection time-out for the first connection attempt is 5000 milliseconds. commThreadCycleInMs : indicates how often data packets are sent back and forth. Reducing this number improves responsiveness, and a default value of 5 is recommended.
Matlab return values	clientID : the client ID, or -1 if the connection to the server was not possible (i.e. a timeout was reached). A call to simxStart should always be followed at the end with a call to simxFinish if simxStart didn't return -1

Source: <http://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.htm>



Seminar Robot-assisted Surgery in Clinics

Surgical robot path planning & simulation based on V-rep

- check if connection is successful

```
if (clientID>-1)
    disp('Connected');
```

- write code for controlling the environment in a structure that is similar to:

```
if (clientID>-1)
    disp('Connected');

    % CODE START

    % CODE END

    % Close connection
    vrep.simxFinish(-1);

end
```

- after the connection is closed, if the interaction tasks are done, delete the vrep object

```
% Delete vrep object
vrep.delete();
```

3. MATLAB side controlling

- MATLAB interacts with V-REP objects through their handles:

```
[returnCode, prismaticJointS22] = vrep.simxGetObjectHandle(clientID, 'Prismatic_joint_S22', vrep.simx_opmode_blocking);
```

simxGetObjectHandle (regular API equivalent: simGetObjectHandle)

Description	Retrieves an object handle based on its name. If the client application is launched from a child script , then you could also let the child script figure out what handle correspond to what objects, and send the handles as additional arguments to the client application during its launch. See also simxGetObjectGroupData .
Matlab synopsis	[number returnCode,number handle]=simxGetObjectHandle(number clientID,string objectName,number operationMode)
Matlab parameters	clientID : the client ID. refer to simxStart . objectName : name of the object. If possible, don't rely on the automatic name adjustment mechanism , and always specify the full object name, including the #: if the object is 'myJoint', specify 'myJoint#', if the object is 'myJoint#0', specify 'myJoint#0', etc. operationMode : a remote API function operation mode . Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	returnCode : a remote API function return code handle : the handle

Source: <http://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.htm>

- get an object position (first function call):

```
[returnCode, needleTipPos] = vrep.simxGetObjectPosition(clientID, needleTip, -1, vrep.simx_opmode_streaming);
```

- get an object position (every call after the first one):

```
[returnCode, needleTipPos] = vrep.simxGetObjectPosition(clientID, needleTip, -1, vrep.simx_opmode_streaming);
```



Seminar Robot-assisted Surgery in Clinics

Surgical robot path planning & simulation based on V-rep

simxGetObjectPosition (regular API equivalent: simGetObjectPosition)

Description	Retrieves the position of an object. See also simxSetObjectPosition , simxGetObjectOrientation and simxGetObjectGroupData .
Matlab synopsis	[number returnCode,array position]=simxGetObjectPosition(number clientID,number objectHandle,number relativeToObjectHandle,number operationMode)
Matlab parameters	clientID : the client ID. refer to simxStart . objectHandle : handle of the object relativeToObjectHandle : indicates relative to which reference frame we want the position. Specify -1 to retrieve the absolute position, sim_handle_parent to retrieve the position relative to the object's parent, or an object handle relative to whose reference frame you want the position operationMode : a remote API function operation mode . Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	returnCode : a remote API function return code position : 3 values representing the position

Source: <http://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.htm>

- set joint position:

```
joint1 = 0.01; % desired position
```

```
[returnCode] = vrep.simxSetJointTargetPosition(clientID, prismaticJointS22, joint1, vrep.simx_opmode_blocking);
```

simxSetJointTargetPosition (regular API equivalent: simSetJointTargetPosition)

Description	Sets the target position of a joint if the joint is in torque/force mode (also make sure that the joint's motor and position control are enabled). See also simxSetJointPosition .
Matlab synopsis	[number returnCode]=simxSetJointTargetPosition(number clientID,number jointHandle,number targetPosition,number operationMode)
Matlab parameters	clientID : the client ID. refer to simxStart . jointHandle : handle of the joint targetPosition : target position of the joint (angular or linear value depending on the joint type) operationMode : a remote API function operation mode . Recommended operation modes for this function are simx_opmode_oneshot or simx_opmode_streaming
Matlab return values	returnCode : a remote API function return code

Source: <http://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.htm>



Seminar Robot-assisted Surgery in Clinics

Surgical robot path planning & simulation based on V-rep

APPENDIX

Example of a minimal MATLAB script

% V-REP Matlab Surgical Robot, bareboneScript.m, 10.05.2017

```
vrep=remApi('remoteApi'); % using the prototype file (remoteApiProto.m)
vrep.simxFinish(-1); % just in case, close all opened connections
clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,5);
```

```
if (clientID>-1)
    disp('Connected');
```

% code

% Handles

```
[returnCode,needleTip] = vrep.simxGetObjectHandle(clientID,'NeedleTip',vrep.simx_opmode_blocking);
```

```
[returnCode,prismaticJointS22] = vrep.simxGetObjectHandle(clientID,'Prismatic_joint_S22',vrep.simx_opmode_blocking);
[returnCode,prismaticJointS21] = vrep.simxGetObjectHandle(clientID,'Prismatic_joint_S21',vrep.simx_opmode_blocking);
[returnCode,prismaticJointS11] = vrep.simxGetObjectHandle(clientID,'Prismatic_joint_S11',vrep.simx_opmode_blocking);
[returnCode,prismaticJointS12] = vrep.simxGetObjectHandle(clientID,'Prismatic_joint_S12',vrep.simx_opmode_blocking);
```

```
[returnCode,prismaticJointS00] = vrep.simxGetObjectHandle(clientID,'Prismatic_joint_S00',vrep.simx_opmode_blocking);
[returnCode,prismaticJointS01] = vrep.simxGetObjectHandle(clientID,'Prismatic_joint_S01',vrep.simx_opmode_blocking);
```

% First read from V-REP of a particular object position

```
[returnCode,needleTipPos] = vrep.simxGetObjectPosition(clientID,needleTip,-1, vrep.simx_opmode_streaming);
```

% Further reads of the same object position

```
[returnCode,needleTipPos] = vrep.simxGetObjectPosition(clientID,needleTip,-1, vrep.simx_opmode_buffer);
```

% Set joint position

joint1 = 0.01; % desired position

```
[returnCode] = vrep.simxSetJointTargetPosition(clientID, prismaticJointS22, joint1, vrep.simx_opmode_blocking);
```

pause(2);

% end code

% Close connection

```
vrep.simxFinish(-1);
```

```
end
```

% Delete vrep object

```
vrep.delete();
```