

## VREP

Description	This is the class constructor and it is responsible for create all the Properties and Methods for the class.
synopsis	<b>Object Class:</b> V = VREP;
Parameters	<b>Library Connection:</b> RemApi (.dll file makes the bridge with Matlab and V-Rep) <b>parameterID:</b> ClientID (ID connection between Matlab and V-REP) <b>parameter:</b> Pioneer object handle (handle numbers). <b>parameter:</b> Objects Any other object you want to handle.
Functions	vConnect; vDisconnect; vGetLaserData; vGetSensorData; vHandle; vObject; vGetObjPosition; vSendControlSignals;
Example	<pre>%% Create the constructor Class  V = VREP;</pre>

## vConnect

Description	This is responsible to connect Matlab and V-Rep simulator.
synopsis	<b>Method:</b> vConnect(vrep);
Parameters	<b>parameterID:</b> ClientID (ID connection between Matlab and V-REP)
return	vrep.clientID (usually it is 1)
Example	<pre>%% Create the constructor Class V = VREP;  %% Connect Matlab to V-Rep V.vConnect;</pre>

## vDisconnect

Description	This is responsible to disconnect Matlab and V-Rep simulator.
synopsis	<b>Method:</b> vDisconnect(vrep);
Parameters	<b>parameterID:</b> ClientID (ID connection between Matlab and V-REP)
return	<b>Int:</b> vrep.clientID (usually it is 0)
Example	<pre>%% Create the constructor Class V = VREP;  %% Connect Matlab to V-Rep V.vConnect;  %% Disconnect Matlab to V-Rep V.vDisconnect;</pre>

## vHandle

Description	This is responsible to get the Pioneer tags on the scene.
synopsis	<b>Method:</b> vHandle(String, Index);
Parameters	<b>String:</b> The name of the Pioneer you want to control on the V-Rep scene. <b>Index:</b> If there is more than one robot, it's necessary to specify the index robot number.
return	<b>Array:</b> vrep.Pioneer(1:19), 1-Pionner3DX, 2- Left Motor, 3- Right Motor, 4:19-Ultrasonic sensor.
Example	<pre> %% Create the constructor Class V = VREP;  %% Connect Matlab to V-Rep V.vConnect;  %% Get handle tags from Pioneer_p3dx V.vHandle('Pioneer_p3dx');  %% Get handle tags from Pioneer_p3dx#1 V.vHandle('Pioneer_p3dx','1');  %% Get handle tags from Pioneer_p3dx#2 V.vHandle('Pioneer_p3dx','2');</pre>

## vGetSensorData

Description	This is responsible to get the position and velocity from Pioneer.
synopsis	<b>Method:</b> vGetSensorData(Index);
Parameters	<b>Index:</b> Usually it is 1, if there is more than one robot, it's necessary to specify the index robot number.
return	<b>Array:</b> Xc(1:3), it returns the cartesian coordinates x, y, and z from the robot center; <b>Array:</b> X(1:3), it returns the cartesian coordinates x, y, and z from the robot control point; <b>Array:</b> U(1:2), it returns the linear and angular velocity from the robot;
Example	<pre> %% Create the constructor Class V = VREP;  %% Connect Matlab to V-Rep V.vConnect;  %% Get handle tags from Pioneer_p3dx V.vHandle('Pioneer_p3dx');  %% Get handle tags from Pioneer_p3dx#1 V.vHandle('Pioneer_p3dx','1');  %% Request position and velocity data from Pioneer_p3dx [Xc1,X1,U1] = V.vGetSensorData(1); %% Request position and velocity data from Pioneer_p3dx#1 [Xc2,X2,U2] = V.vGetSensorData(2);</pre>

## vObject

Description	This is responsible to get the tag from any other object on the scene.
synopsis	<b>Method:</b> vObject(String);
Parameters	<b>String:</b> The object's name you want to control on the V-Rep scene.
return	<b>Int:</b> vrep.Object{ }, with the object tag.
Example	<pre>%% Create the constructor Class V = VREP;  %% Connect Matlab to V-Rep V.vConnect;  %% Get handle tags from object Disc V.vObject('Disc');</pre>

## vGetObjPosition

Description	This is responsible to get the position and orientation from any other object on the scene.
synopsis	<b>Method:</b> vGetObjPosition(String);
Parameters	<b>String:</b> The object's name you want to request data on the V-Rep scene.
return	<b>Array:</b> Xc(1:3), it returns the cartesian coordinates x, y, and z from the object center; <b>Array:</b> A(1:3), it returns the angles from axis x, y and z.
Example	<pre>%% Create the constructor Class V = VREP;  %% Connect Matlab to V-Rep V.vConnect;  %% Get handle tags from object Disc V.vObject('Disc');  %% Get position and angle from object Disc [Xc, A]= V.vGetObjPosition('Disc');</pre>

## vGetLaserData

Description	This is responsible to get the laser data from Pioneer.
synopsis	<b>Method:</b> vGetLaserData(Index);
Parameters	<b>Index:</b> Usually it is 1, if there is more than one robot, it's necessary to specify the index robot number.
return	<b>Array:</b> Map(1:181,1:3), the first column indicates the coordinate x values, the second column indicates the y values, and the third column indicates the distance.
Example	<pre>%% Create the constructor Class V = VREP;  %% Connect Matlab to V-Rep V.vConnect;  %% Get handle tags from Pioneer V.vHandle('Pioneer_p3dx');  %% Get laser data Map = V.vGetLaserData(1);</pre>

## vSendControlSignals

Description	This is responsible to send control signals to Pioneer.
synopsis	<b>Method:</b> vSendControlSignals(Velocity,Index);
Parameters	<b>Velocity:</b> It contains on the first index the linear velocity, and on the second index the angular velocity. <b>Index:</b> Usually it is 1, if there is more than one robot, it's necessary to specify the index robot number.
return	<b>None</b>
Example	<pre>%% Create the constructor Class V = VREP;  %% Connect Matlab to V-Rep V.vConnect;  %% Get handle tags from Pioneer V.vHandle('Pioneer_p3dx');  %% Set Velocity Ud = [0.3;0]; V.vSendControlSignals(Ud,1);</pre>