

V-REP MATLAB használata

Előkészület

Ahhoz, hogy a MATLAB segítségével elérjük a V-REP szimulációs környezetet, az alábbi fájlokat kell átmásolnunk a munkakönyvtárunkba.

- **simpleTest.m**: ezzel tesztelhetjük, hogy minden megfelelően működik.
- **remApi.m**: a szükséges MATLAB interfészt tartalmazza.
- **remoteApi.dll**: a könyvtár, amivel elérjük majd a VREP-et.

A V-REP alaphelyzetben folyamatos **RemoteAPI** kapcsolatot indít, induláskor azonban jelzi, ha nem sikerül elindulnia. Ha ezeket átmásoltuk egy mappába, indítsuk el MATLAB-ot és a V-REP-et. Futtassuk a **simpleTest.m** fájlt. Ha minden jól működik a V-REP-ből visszakapjuk az egér pozícióját.

Ha nem működik, több dolgot kell ellenőriznünk. Nézzük meg a **simpleTest.m** fájlban, a csatlakozás utasítását.

```
clientID=vrep.simxStart('127.0.0.1',19997,true,true,5000,5);
```

Sorrendben tehát az IP-cím, port. Ezeket kell összehasonlítani a V-REP telepítési mappájában található **remoteApiConnections.txt** fájl tartalmával. Amennyiben más a portszám vagy az IP-cím, a MATLAB fájlban változtassuk meg.

Szimuláció MATLAB

Csatlakozás – első lépések

Győződjünk meg róla, hogy a megfelelő módon elérhető a **RemoteAPI** könyvtár és a **remApi.m** fájl. A következő MATLAB parancs segítségével inicializáljuk a VREP klienst.

```
vrep=remApi('remoteApi');
```

Állítsunk le minden korábbi kapcsolatot.

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

Végül csatlakozzunk MATLAB-al.

```
clientID=vrep.simxStart('127.0.0.1',19997,true,true,5000,5);
```

Ezt függvénybe is szervezhetjük, az alábbiak szerint.

```
function [vrep,clientID] = connectVREP(ip,port)
    vrep=remApi('remoteApi');
    vrep.simxFinish(-1);
    clientID=vrep.simxStart(ip,port,true,true,5000,5);
end
```

Ezután a VREP megfelelő funkcióit tudjuk majd használni.

Pozíció lekérdezés

Találjuk meg azt a részkomponenst, ami le akarunk kérdezni. Például, ha betöltjük a **testJKK** fájlt, látjuk a háromkerekű robotunkat. Ehhez hozzá lett rakva egy lézerszenzor az elejére.

Ahhoz, hogy a VREP műveleteket tudjon végezni a robotunkkal, le kell kérdeznünk az object handler-jét. Találjuk meg a robot nevét V-REP-ben majd megadjuk az eszerinti végrehajtást:

```
[err,result] = vrep.simxGetObjectHandle(clientID, 'Pioneer_p3dx',
vrep.simx_opmode_oneshot_wait)
```

Ezután lekérdezhető az objektum pozíciója, felhasználva a megkapott object handler-t.

```
[err,pos] = vrep.simxGetObjectPosition(clientID,result,-1,vrep.simx_opmode_oneshot_wait)
```

Hasonlóan a robot orientációját.

```
[err,ori] = vrep.simxGetObjectOrientation(clientID, jointLBR, -1, vrep.simx_opmode_oneshot_wait)
```

Fontos, hogy mindkét esetben a lekérdezés módja várakoztatás kell hogy legyen

(**simx_opmode_oneshot_wait**), mivel addig nem akarunk továbblépni, ameddig nem kapunk vissza valamilyen üzenetet.

Lézerszkennerek adat kiolvasása

Hasonló logikával, mint az előbb találjuk meg a lézerszkennerek nevét és kérdezzük le a kezelő konstansát (object handler):

```
[err,sensor] = vrep.simxGetObjectHandle(clientID, 'LaserScannerLaser_2D', vrep.simx_opmode_oneshot_wait)
```

Ezután olvassuk ki az adatokat:

```
[err,result,poi,i,v] = vrep.simxReadProximitySensor(clientID, sensor, vrep.simx_opmode_oneshot_wait)
```

Az eredményként megkapjuk a legkisebb mért távolságot, a detektált pontok számát és a detektált felület normálvektorát.

Motor mozgatása

Válasszunk ki egy csuklót és annak jelöljük ki új pozíciót. A példában én a robot jobb kerekét jelöltem ki.

```
[err,rw] = vrep.simxGetObjectHandle(clientID, 'Pioneer_p3dx_rightMotor', vrep.simx_opmode_oneshot_wait)
```

A visszatért értéket felhasználhatjuk a továbbiakban, ha új célsebességet akarunk kijelölni:

```
vrep.simxSetJointTargetVelocity(clientID, rw, 1, vrep.simx_opmode_oneshot_wait)
```

Ezt a logikát felhasználva már könnyű előre küldeni a robotot. Csúpnak le kell kérdezni a másik motort és ugyanezt végrehajtani.

Példa: Egyszerű falkerülő mechanizmus

```
% A simple controller to avoid obstacles
% Motor encoder incrementation
%

laser_threshold = 2.0
laser_threshold_sup = 9.0
% Get object handler for both motors
[err,rw] = vrep.simxGetObjectHandle(clientID, 'Pioneer_p3dx_rightMotor',...
    vrep.simx_opmode_oneshot_wait)
[err,lw] = vrep.simxGetObjectHandle(clientID, 'Pioneer_p3dx_leftMotor',...
    vrep.simx_opmode_oneshot_wait)
% Get handle for laser scanner
[err,sensor] = vrep.simxGetObjectHandle(clientID, 'LaserScannerLaser_2D',...
    vrep.simx_opmode_oneshot_wait)
velocityMagnitude = 3.0
targetvelocityR = velocityMagnitude;
targetvelocityL = targetvelocityR;
vrep.simxSetJointTargetVelocity(clientID, rw, targetvelocityR,...
    vrep.simx_opmode_oneshot_wait)
vrep.simxSetJointTargetVelocity(clientID, lw, targetvelocityL,...
```

```

    vrep.simx_opmode_oneshot_wait)
% The algorithm itself
while 1
    [err,result,poi,i,v] =...
        vrep.simxReadProximitySensor(clientID, sensor, ...
            vrep.simx_opmode_oneshot_wait);
    if result && laser_threshold_sup > norm(poi)
        if (norm(poi) < laser_threshold)
            if targetvelocityR > -velocityMagnitude
                targetvelocityR = targetvelocityR...
                    - velocityMagnitude*sigmoid_simple(1/(norm(poi)*1000));
            end
        else
            if targetvelocityR < velocityMagnitude
                targetvelocityR = targetvelocityR...
                    + velocityMagnitude*sigmoid_simple(1/(norm(poi)*1000));
            end
        end
        vrep.simxSetJointTargetVelocity(clientID, rw, targetvelocityR,...
            vrep.simx_opmode_oneshot_wait);
    else
        targetvelocityR = velocityMagnitude;
    end
end
end

```

Kép kinyerése kamerából

Lehetőség van kamerakép kinyerésére is. Tekintsük a következő esetet, amikor egy kinect-et szereltünk a robotra. Kérdezzük le az objektum azonosítóját.

```

[err,kinect] = vrep.simxGetObjectHandle(clientID, 'kinect_rgb',
vrep.simx_opmode_oneshot_wait)

```

Ezzel lekérdeztük a kinect kamerájának elérését. Ahhoz hogy adatokat nyerjünk ki, kérdezzük le a szenzor adatait:

```

[err,i,k] = vrep.simxGetVisionSensorImage2(clientID,kinect,0,vrep.simx_opmode_oneshot_wait)

```

Ha a harmadik paramétert 1-re állítjuk szürkeárnyaltos képet kapunk, amivel sok esetben sokkal könnyebb dolgozni.

Kapcsolat bontása

Mintha egy sima hálózati kapcsolatot szeretnénk bontani. Először lekérdezzük mekkora volt a ping idő, lebontjuk a kapcsolatot, majd megsemmisítjük az adott VREP referenciát. Szükséges ezeket a lépéseket elvégezni, ha el szeretnénk kerülni, hogy a MATLAB hibával lépjen ki. Emellett ajánlott egy funkciót készíteni, hiszen ezt sok esetben szeretnénk használni, akár különböző paraméterekre is.

A megfelelő MATLAB funkció:

```

function disconnectVREP(vrep, clientID)
    vrep.simxGetPingTime(clientID);
    vrep.simxFinish(clientID);
    vrep.delete();
end

```

Készítette: Hajdu Csaba