```
clear all; close all; clc;

% UPD Multicasting received packets
% create UDP multicast socket
port = 5412;
group = "230.8.6.7";
sock = udpport("LocalPort", port, "EnablePortSharing", true);
configureMulticast(sock, group);
hoursEnd = 3; % SET SET hours to kill the process after
switch exist("hoursEnd", "var")
    case 0
        minEnd = 120; % SET SET min to kill the process after (2 hrs)
    case 1

        minEnd = hoursEnd*60; % if hoursEnd is set
end

timeElap = 0; % time elapsed init

% Rigidbody data streaming related stuff
addpath(genpath(['C:\Users\Hari\Documents\GitHub\NatNet_MATLAB_Streaming' ...
    '\NatNet_SDK_4.1\NatNetSDK\Samples\Matlab\'])); % add the samples folder
dllPath = fullfile('C:','Users','Hari','Documents/GitHub/NatNet_MATLAB_Streaming/', ...
    'NatNet_SDK_4.1/NatNetSDK/lib/x64/NatNetML.dll');
assemblyInfo = NET.addAssembly(dllPath);
fprintf( 'NatNet Polling Sample Start\n' )
```

```
NatNet Polling Sample Start
```

```

% create an instance of the natnet client class
fprintf( 'Creating natnet class object\n' )
```

```
Creating natnet class object
```

```
natnetclient = natnet;

% connect the client to the server (multicast over local loopback) -
% modify for your network
fprintf( 'Connecting to the server\n' )
```

```
Connecting to the server
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% natnetclient.HostIP = '10.0.0.3'; % MoCap Computer
% natnetclient.ClientIP = '10.0.0.68'; % Hari's Laptop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
natnetclient.HostIP = '127.0.0.1';
natnetclient.ClientIP = '127.0.0.1'; % MoCap Computer Loopback
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
natnetclient.ConnectionType = 'Multicast';
natnetclient.connect;
if ( natnetclient.IsConnected == 0 )
        fprintf( 'Client failed to connect\n' )
        fprintf( '\tMake sure the host is connected to the network\n' )
        fprintf( '\tand that the host and client IP addresses are correct\n\n' )
        return
end

% get the asset descriptions for the asset names
model = natnetclient.getModelDescription;

% get the pause time period based on the framerate
pauseTime = 1/natnetclient.FrameRate; % actual framerate
% pauseTime = 0.1; % 10Hz lower framerate

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Poll for the rigid body data at framerate being multicast
fprintf( '\tStarting UPD multicast stream of position data!\n\n')
```

```
        Starting UPD multicast stream of position data!
```

```
% write packet stuff
tic
while timeElap < 60*minEnd
    pause( pauseTime );
    data = natnetclient.getFrame;
```

```
    try
        writeData = num2str([ data.RigidBodies( 1 ).x, ...
                              data.RigidBodies( 1 ).y, ...
                              data.RigidBodies( 1 ).z, ...
                              EulerAngles(...
                              quaternion(data.RigidBodies(1).qx,...
                              data.RigidBodies(1).qy,...
                              data.RigidBodies(1).qz,...
                              data.RigidBodies(1).qw), 'zyx')']); % try sending the position data: [x, y, z, 3_attitudes]
    catch
        continue;
    end
    write(sock, writeData, group, port);
    timeElap = toc;
end

% destroy socket
configureMulticast(sock, "off")
clear sock
fprintf( '\tDone streaming!\n\n')
```

```
	Done streaming!
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% testing space
tic
q = quaternion(data.RigidBodies(1).qx, data.RigidBodies(1).qy, data.RigidBodies(1).qz, data.RigidBodies(1).qw)
Ang = EulerAngles(q, 'zyx')
toc


tic
Ang = EulerAngles(quaternion(data.RigidBodies(1).qx, data.RigidBodies(1).qy, data.RigidBodies(1).qz, data.RigidBodies(1).qw), 'zyx')'
toc
```