**PNO\_data.m**

clear all

clc

for i=1:1:100

%%cell2mat: Converts readings from fastrak to an array

%%fstclient(host string name, port number, time of data collection (s))

table=cell2mat(fstclient('127.0.0.1',7234,0.1));

%Takes last row of table, as this is most recent values (x,y,z)

Sensor1 = table(end,:)

%For four sensors:

% Sensor1 = table(end-3,:)

% Sensor2 = table(end-2,:)

% Sensor3 = table(end-1,:)

% Sensor4 = table(end,:)

% % % InitialSensor1 = load('PNO\_initial\_1.txt')

Sensor1 = double(Sensor1);

save('PNO\_data1.txt','Sensor1','-ascii')

type('PNO\_data1.txt')

%For four sensors:

% Sensor2 = double(Sensor2)

% save('PNO\_data2.txt','Sensor2','-ascii')

% type('PNO\_data2.txt')

% Sensor3 = double(Sensor3)

% save('PNO\_data3.txt','Sensor3','-ascii')

% type('PNO\_data3.txt')

% Sensor4 = double(Sensor4)

% save('PNO\_data4.txt','Sensor4','-ascii')

% type('PNO\_data4.txt')

end

**Fstclient.m**

used to connect to fserver and parse frames, assumes default output

% records and that the tracker is a Fastrak (any version)

function pno\_data = fstclient(host\_str, port\_num, seconds)

% use java Socket and DataInputStream classes

import java.net.Socket

import java.io.\*

% connect to the socket and open the data stream

for attempt = 1:5

try

fprintf(1, 'Attempt #%d to connect...', attempt);

socket = Socket(host\_str, port\_num);

stream = socket.getInputStream;

di\_stream = DataInputStream(stream);

fprintf(1, 'Success\n');

break;

catch

fprintf(1, 'Failure\n');

if ~isempty(socket)

socket.close;

end

if attempt == 5

error('Failed all attempts to connect');

end

pause(1);

end % try

end % for

pno\_data = {};

% collect data and store it in the array

t1 = tic;

while toc(t1) < seconds

try

% skip #bytes in header to station number

tmp = zeros(1,1);

di\_stream.read(tmp,0,1);

% get station number

snum = single(di\_stream.readByte) - 48;

% skip more of the header

tmp = zeros(1,1);

di\_stream.read(tmp,0,1);

% get pno data

pno = zeros(1,6,'single');

for i = 1:6

pno(i) = swapbytes(single(di\_stream.readFloat) );

end

% skip crlf

tmp = zeros(1,2);

di\_stream.read(tmp,0,2);

% create row with data

%%following line commented to exclude rotational degrees of

%%freedom

%pno\_row = {snum pno(1) pno(2) pno(3) pno(4) pno(5) pno(6)};

pno\_row = {pno(2)}; %only x

% append row to pno\_data

pno\_data = cat(1,pno\_data,pno\_row);

catch err

fprintf(1, err.message() );

break;

end

end

% disconnect from the socket

fprintf(1, 'Disconnected...\n');

socket.close();

end

**PNO\_initial.m**

clear all

clc

%for loop not necessary to read initial reference point for each sensor

table=cell2mat(fstclient('127.0.0.1',7234,0.1));

%Takes last row of table, as this is most recent values (x,y,z)

initialSensor1 = table(end,:)

%For four sensors:

% initialSensor1 = table(end-3,:)

% initialSensor2 = table(end-2,:)

% initialSensor3 = table(end-1,:)

% initialSensor4 = table(end,:)

initialSensor1 = double(initialSensor1);

save('PNO\_initial\_1.txt','initialSensor1','-ascii')

type('PNO\_initial\_1.txt')

%For four sensors:

% initialSensor2 = double(initialSensor2)

% save('PNO\_initial\_2.txt','initialSensor2','-ascii')

% type('PNO\_initial\_2.txt')

% initialSensor3 = double(initialSensor3)

% save('PNO\_initial\_3.txt','initialSensor3','-ascii')

% type('PNO\_initial\_3.txt')

% initialSensor4 = double(initialSensor4)

% save('PNO\_initial\_4.txt','initialSensor4','-ascii')

% type('PNO\_initial\_4.txt')