MODELING AND SIMULATION OF THE SENSING AND COMMUNICATION OF REAL WORLD PHENOMENA USING NETSIM WSN

**Objective:** To model real physical parameters such as Temperature, Pressure, Humidity etc. and to simulate their sensing and communication using NetSim WSN model

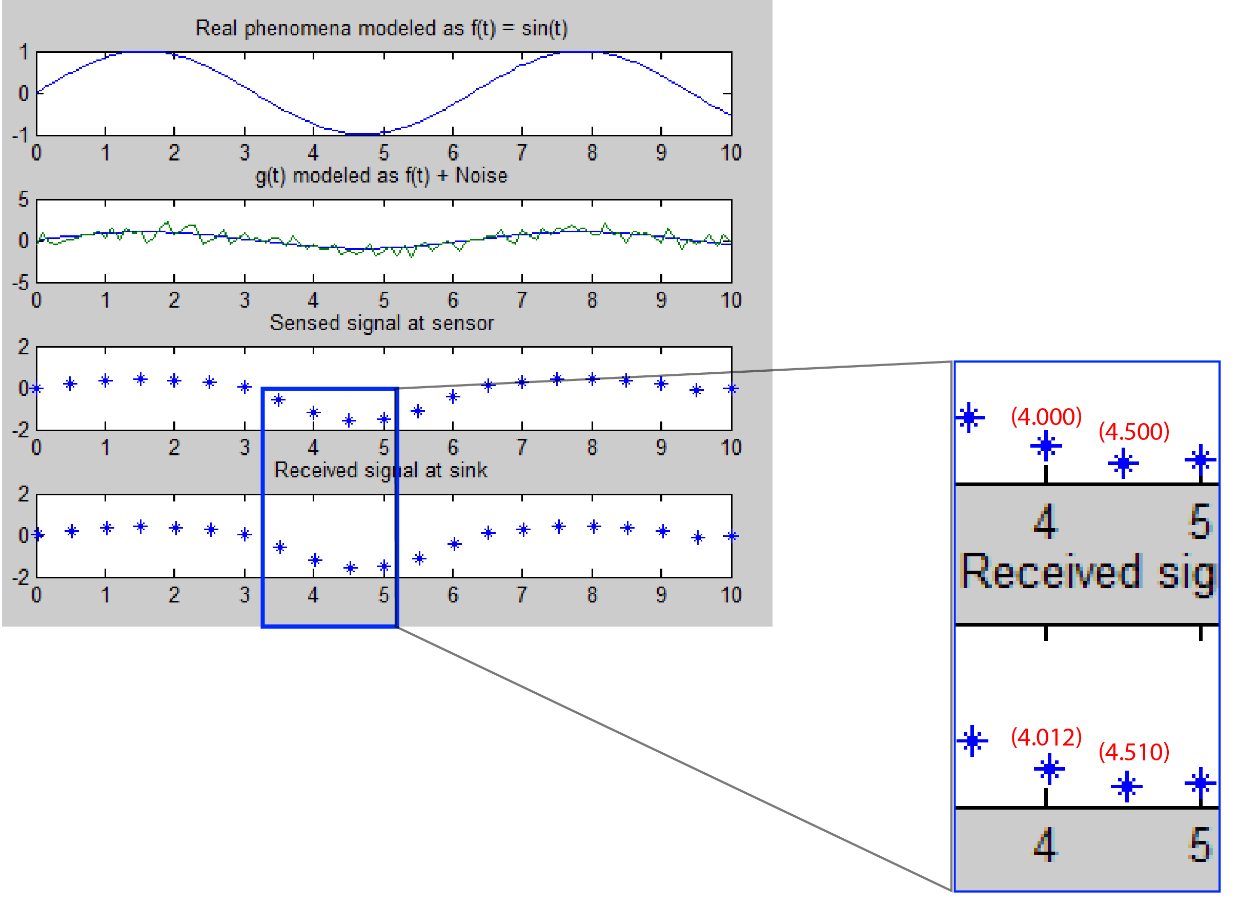
Let the variation of any physical parameter over time defined by f (t).

Due to a variety of effects in the real world, this signal would appear to the sensor as from g (t) which we define as f (t) plus error (which we model as Additive White Gaussian Noise)

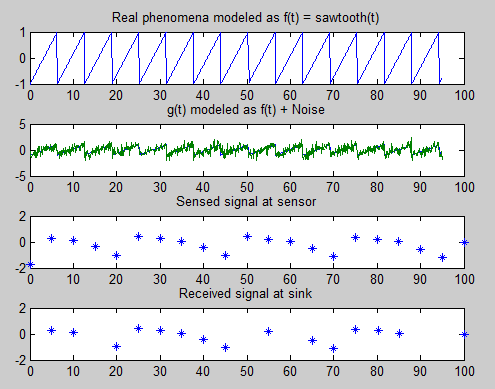
**NetSim-MATLAB Interfacing:** NetSim provides an easy way of interfacing with MATLAB. Given below is what g (t) and f (t) would look like.

The following graphs depict the model of a real world phenomena varying over time, the additional of noise to this, the sensing of this signal by the senor, and then the transmission (after possible errors and delay) to the sink

**Simulation of network communication using NetSim**: While the signal is modelled in MATLAB, NetSim simulates the 802.15.4 protocol in MAC and PHY layers. This simulation determines the time taken for the packet to reach the sink, and also whether the packet is errored or not. Users can also modify the underlying source code of NetSim.

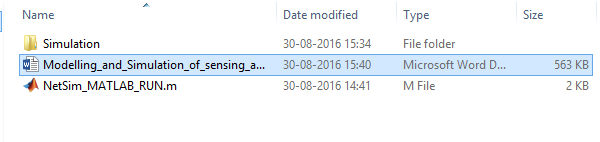


If we increase the path loss, some of the packets will get errored. If you observe the below graph, some of the packets are errored at time stamps (in seconds) - 15, 50, 60, 90, and 95.

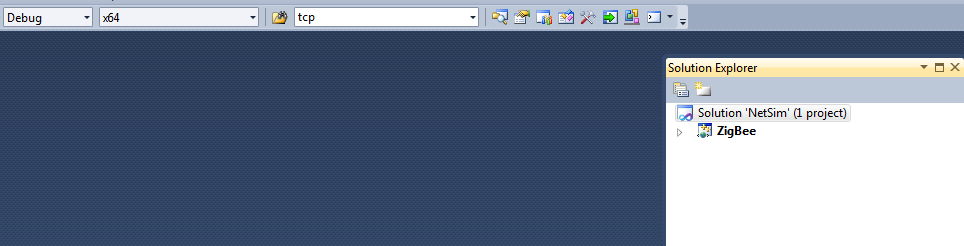


**How to run this project in NetSim:**

1. After you unzip the file the folder would look like



1. Open the simulation folder
2. Open the NetSim.sln file inside the Simulation folder, it displays a message “One or more projects in the solution were not loaded completely”
3. Click on OK and when this opens in MS Visual Studio 2010, it would look like
4. Copy the NetSim\_MATLAB\_RUN.m file and paste this in the path where MATLAB is installed.



For example “C:\Program Files (x86)\MATLAB\R2011a”.The NetSim\_MATLAB\_RUN.m contains the following lines of code

MATLAB Code

function WSN = Nevent\_timeSim\_MATLAB\_RUN(simulation\_time, event\_time,current\_time,flag)

clc;

close;

rng(1) ;

if (flag==0)%sensor is sensing the signal

if (event\_time==0)

t=(0:0.1:simulation\_time)';

x = sawtooth(t);% This is the sawtooth signal f(t)

y = awgn(x,1,'measured');% This is additive white gaussian noise

end

%f(t) and noise at every event

x1 = sawtooth(event\_time);

y1 = awgn(x1,1,'measured');

a = y1;

disp(a);

%Copying event time and noise to a file

fileID1 = fopen('sensor.txt','A');

fprintf(fileID1,'%f %f\r\n',event\_time,a);

fclose (fileID1);

WSN = a;

end

if(flag == 1)%signal received at sink

x2=sawtooth(current\_time);

y2=awgn(x2,1,'measured');

%Copying current time and noise to a file

fileID2 = fopen('sink.txt','A');

fprintf(fileID2,'%f %f\r\n',current\_time,y2);

fclose (fileID2);

end

if (flag == 2)

fileID1 = fopen('sensor.txt','A');

fprintf(fileID1,'%f %f\r\n',simulation\_time,0);

fclose (fileID1);

fileID1 = fopen('sink.txt','A');

fprintf(fileID1,'%f %f\r\n',simulation\_time,0);

fclose (fileID1);

%Reading values from files

fileID1 = fopen('sensor.txt');

sensor = textscan(fileID1,'%f %f');

fclose(fileID1);

fileID2 = fopen('sink.txt');

sink = textscan(fileID2,'%f %f');

fclose(fileID2);

%Graphs plotting

figure

t=(0:0.1:simulation\_time)';

x = sawtooth(t);

y = awgn(x,1,'measured');

subplot(4,1,1)

plot(t,x)

title('Real phenomena modeled as f(t) = sawtooth(t)')

subplot(4,1,2)

plot(t,[x y])

title('g(t) modeled as f(t) + Noise')

subplot(4,1,3)

plot(sensor{1},sensor{2},'\*')

title('Sensed signal at sensor')

subplot(4,1,4)

plot(sink{1},sink{2},'\*')

title('Received signal at sink')

delete('sensor.txt');

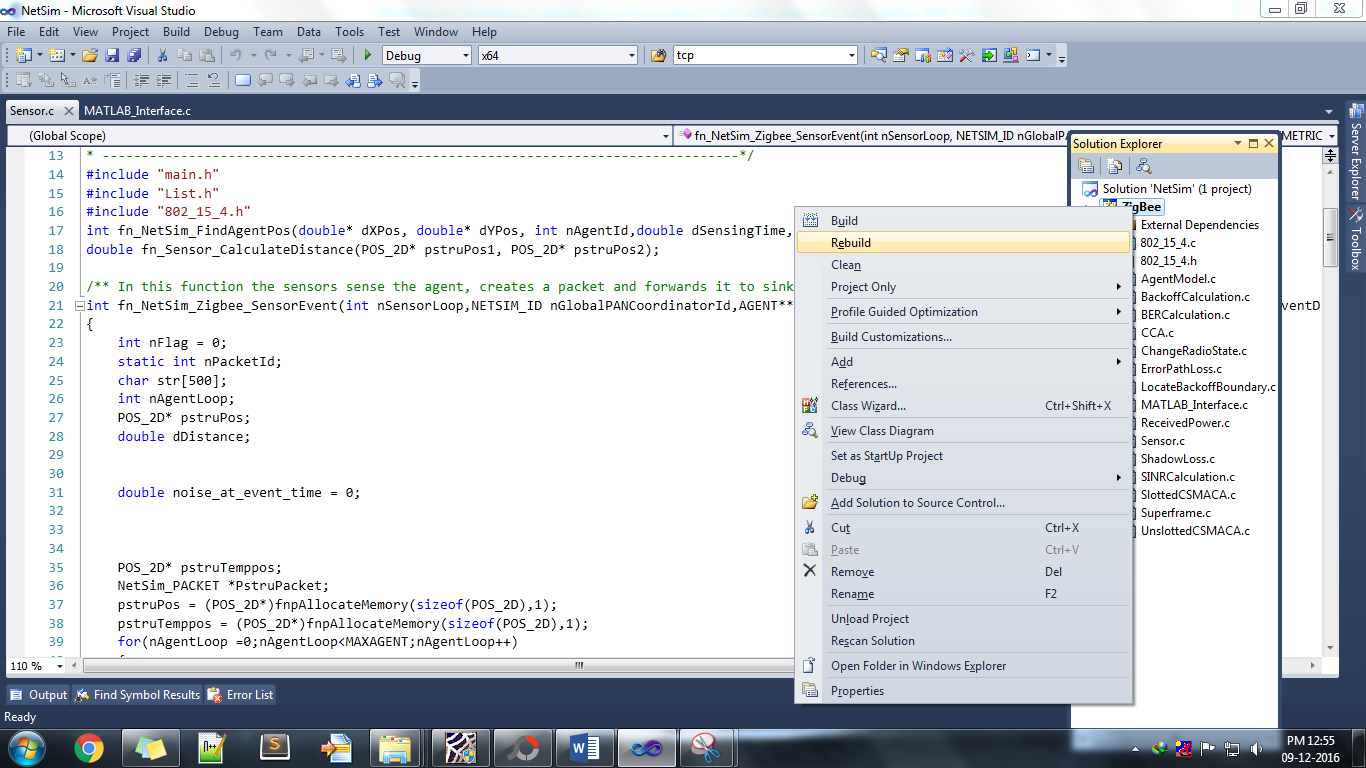
delete('sink.txt');

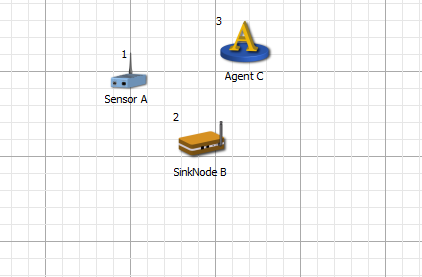
end

end

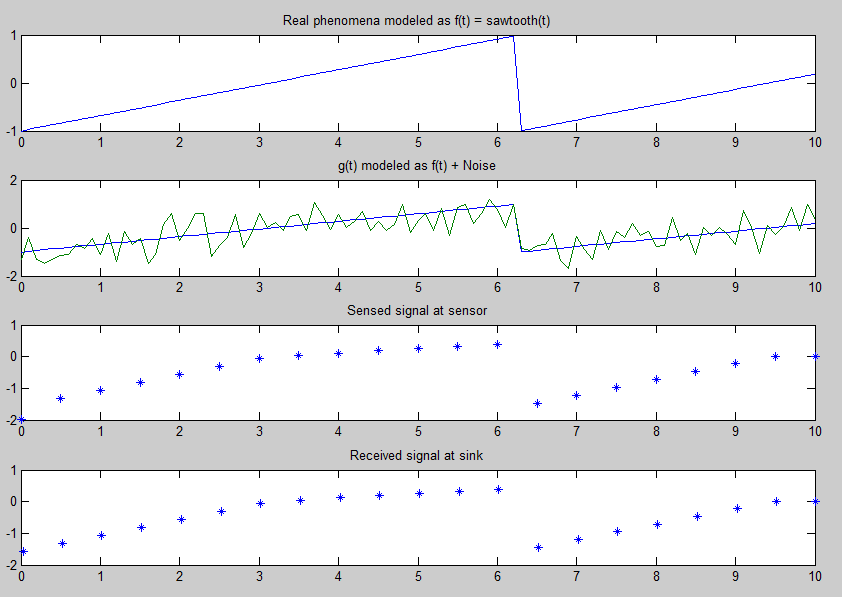
1. To compile a MATLAB engine application in the Microsoft Visual Studio 10.0 (2010) environment, follow the steps 12,13,14 present in 8.4.1 section of NetSim User Manual

**NOTE:** If the above mentioned (step 6) settings are already there, please replace it

1. Right click on Zigbee in Solution Explorer and select rebuild.
2. Upon rebuilding, libZigBee.dll will get created in the path ..\simulation\DLL
3. Now copy the libZigBee.dll from this DLL folder and paste it in NetSim bin folder present in the NetSim installation directory. The NetSim install director would look something like < C:\Program Files (x86)\NetSim Standard\bin>
4. Note that there exists a libZigbee.dll in this bin folder. This is the default file being shipped with NetSim. The user is replacing this file with the newly built file.
5. Therefore, take care to rename the original libZigBee.dll file, so that it isn’t lost. For example, you may rename it as libZigbee\_default.dll
6. Run NetSim in administrator mode and create a Network Scenario in NetSim WSN Network with 1 sensor, 1 sinknode and 1 agent



1. Run the simulation
2. Now you can see the different plots in MATLAB like Real phenomena f (t), f (t) + Noise, Sensed signal at sensor and the received signal at sink.



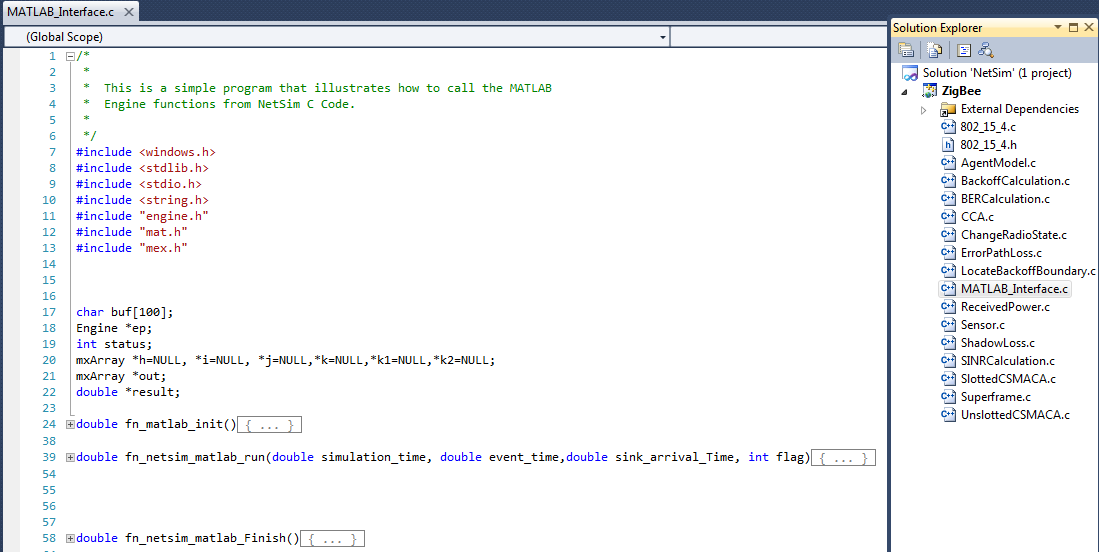
**Code modifications done:**

Please note that this projects contains a MATLAB\_Interface.c file which contains the following functions

a) fn\_netsim\_matlab\_init() - Opens the MATLAB Engine

b) fn\_netsim\_matlab\_run() - Communicates with MATLAB Command Window

c) fn\_netsim\_matlab\_finish() - Closes the MATLAB Engine



Then we have added the following lines of code in fn\_NetSim\_Zigbee\_SensorEvent()function present in Sensor.c file

#ifdef \_SENSOR\_INFORMATION

noise\_at\_event\_time = get\_sensor\_info(PstruPacket);

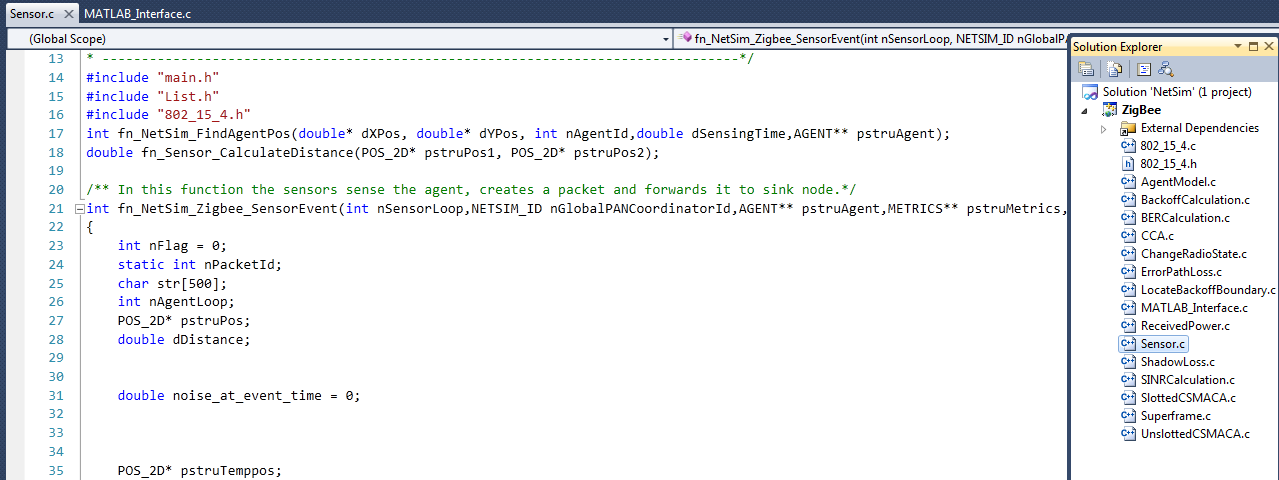
#else

sprintf(str,"%lf",noise\_at\_event\_time);

PstruPacket->szOverhead = \_strdup(str);

#endif

We have then initialised the variable double noise\_at\_event\_time = 0; in fn\_NetSim\_Zigbee\_SensorEvent()function present in Sensor.c file



Then we have added the following lines of code in fn\_NetSim\_ZigBee\_SinkNodeApp() function present in 802.15.4.c

#ifdef \_SINKNODE\_INFORMATION

get\_sinknode\_info(pstruEventDetails->pPacket);

#endif

Then we have added the following lines of code in fn\_NetSim\_Zigbee\_Init() function present in 802.15.4.c

fprintf(stderr, "Initializing MATLAB\n");

fn\_matlab\_init();

We have added the following lines of code in 802.15.4.h

double fn\_matlab\_init();

double fn\_netsim\_matlab\_run(double simulation\_time, double event\_time, double sensing\_interval, int count,double sink\_arrival\_Time,int flag);

double fn\_netsim\_matlab\_Finish();

#define \_SENSOR\_INFORMATION

#define get\_sensor\_info(pstrupacket) fn\_netsim\_matlab\_run((NETWORK->pstruSimulationParameter->dVal)/1000000,(pstruEventDetails->dEventTime)/1000000, (pstru\_Sensor[nSensorLoop]->dSensingInterval)/1000000,0,0,0)

#define \_SINKNODE\_INFORMATION

#define get\_sinknode\_info(pstrupacket) fn\_netsim\_matlab\_run((NETWORK->pstruSimulationParameter->dVal)/1000000,0, 0, count,pstruEventDetails->dEventTime/1000000, 1)

Then we have added the following lines of code in fn\_NetSim\_Zigbee\_Finish() function present in 802.1504.c

fn\_netsim\_matlab\_run((NETWORK->pstruSimulationParameter->dVal)/1000000,0,0,2);

fprintf(stderr,"Press any key to close MATLAB\n");

getch();

fn\_netsim\_matlab\_Finish();

**Possible Future Development**

1. In this example the physical phenomena has been modelled as f (t). This can be extended as f (x, y, z, t) which takes into account space and time