Enhancing Learning Using Modular Wireless Sensor Networking (WSN) Hands-On Experiments

Ezgi Taslidere[1], Fernand S. Cohen[1], and Fredricka Reisman[2]

[1] Department of Electrical and Computer Engineering
[2] School of Education and Goodwin College of Professional Studies
DREXEL UNIVERSITY, Philadelphia, PA 19104, USA
{ezgi,fscohen,reismafk}@drexel.edu

Abstract. This paper presents the use of WSN in educational research as a platform for enhanced learning through hands-on modular experiments to illustrate abstract theoretical concepts in diverse courses in Electrical Engineering. The WSN consists of Mica2 motes with on-board sensors, wireless communication antennas, and processors that are programmed using NesC. Three sets of experiments feeding into different courses (e.g., wireless embedded networks, detection and estimation, stochastic processes, probability theory, statistical pattern recognition, and digital signal processing) and illustrating different theoretical concepts are presented in details. These experiments can be used as demos in those courses and/or can be incorporated as hands-on laboratory projects to go hand in hand with the course. Assessment of the experiments as pedagogical tools are also presented through well designed evaluation questionnaires given to the students. Assessment survey shows that both the sensor network platform and the novel experiments built on them are pedagogically successful tools.

Keywords: Wireless sensor networks, educational research, target detection, receiver operating characteristics (ROC) curve, decision making, and motion tracking.

1 Introduction

Translating abstract theory into real life application problems is extremely important in engineering education. Most of the time it is difficult for students to link advanced theoretical concepts in courses to real life applications. Hence bridging this gap between theory and applications is highly needed, timely and very desirable. Towards that end we integrate a practical approach to assist the learning and understanding of various theoretical concepts in a wide range of Electrical and Computer Engineering courses through the use of WSN.

WSN is an emerging new field that draws applications from diverse disciplines in engineering and science and stands as an excellent candidate to be used as a platform for illustrating abstract theoretical concepts in diverse courses in electrical and computer engineering curricula.

J. Zheng et al. (Eds.): ADHOCNETS 2009, LNICST 28, pp. 506–522, 2010.

There are various institutions working on incorporating applied wireless sensor networks in courses on embedded systems [2-11]. Our work complements these activities but it also differs from them in the use of WSN platform as a pedagogical tool, to teach students advanced-level theoretical concepts in a variety of electrical engineering courses. Through WSN judiciously designed hands-on experiments, these concepts are easier for the students to grasp, and make these concepts more tangible and applied. We have used WSN in a variety of courses outside of the networking area such as digital signal processing, stochastic and non-deterministic systems, pattern recognition, and computer vision. We have also used our modular experiments in a course on embedded system. The modularity of the experiments makes them portable and adaptable to a wide range of electrical and computer engineering courses. We have targeted undergraduate students at the junior and senior levels, as well as graduate students. The overarching objective here is not to develop new course material but rather use innovative hands-on experiments to illustrate some of the theoretical concepts in these courses, to flash out the applicability of these concepts to real life problems, and to finally make the teaching and learning of these concepts friendlier and fun.

We report on a set of 3 experiments as testbeds to show how theory and practice are wedded and linked. While the first experiment is geared towards illustrating concepts in nodes communication and sensing; the other two are more modular and are geared towards non specialized network courses. The impact on enhancing learning is assessed through a carefully design questionnaire and the analysis of that output and findings is carried on by our in house expert collaborator in learning assessment. The first experiment is called "Basic Target Detection" and designed to be used in a WSN Lab course in order to show students the basic concepts of communication between nodes and a cluster node as well as sync. The second experiment called "Local, Selected and Consensus Decision on Target Detection" is more advanced than the first one and is designed to illustrate concepts of detection and decision making in advanced courses dealing with detection and estimation theory, such as Statistical Engineering theory, Statistical Pattern Recognition and Digital Signal Processing. The third experiment called "Motion Tracking" includes various concepts from detection and estimation theory, motion estimation and dynamic updates, and has been used in our signal processing course, computer vision, detection and estimation and pattern recognition course.

The paper is organized as follows. Designed experiments are presented in Section II. The Assessment Survey and the survey results are presented in Section III. Our conclusions on the overall results/evaluations are given in Section IV.

2 Wireless Sensor Networks Platform Experiments

The WSN platform with its various hardware components used for our experiments is shown in Figure 1 (MIB510 Programming Board, MICA2 Mote, MTS300 Sensor Board) [12]. These products are from Crossbow and are used by various schools and commercial establishments [2-7]. Matlab [13] was used for designing the graphical user interface in those experiments.

Theoretical Concepts Covered in the Three Experiments

The theoretical concepts covered in the test bed experiments are: 1. Basic concepts of target detection; 2. Concepts of communication between nodes and a cluster node; 3. Empirical histograms as estimates for probability density functions; 4. Concepts of binary hypotheses; 5. False alarm rates (type 1 error) and misdetection (type 2 error); 6. Receiver operating characteristic (ROC) curve (probability of detection versus false alarm curve) for two class distributions; 7. Likelihood ratio detectors (Neyman Pearson (NP) and Bayes Classifiers) test classifiers and their relationship to ROC curves; 8. Concepts of Motion estimation and dynamic updates.

2.1 Experiment 1 – Basic Target Detection

Where Would the Experiment Fit?

This experiment has two uses depending on the course content, student knowledge, and time allocated to it. Firstly, it can be a final lab exercise or a final project in embedded network course. Here at Drexel, it was made part of the junior level lab course on wireless embedded networks (see syllabus in Appendix). This course introduces students to wireless sensor networks and to their architecture, and provides them with the opportunity to gain hands-on experience with designing and programming the motes and seeing their use in WSN applications. This course was designed by our collaborator on the NSF educational grant supporting this activity [see Appendix]. Our experiment would fit in the final part of this course, as a project that will require them to apply the knowledge gained by the students in programming the motes and have to communicate with one another with a real application as target detection using the mote sensors to detect a target in the vicinity of the motes and to communicate that information to a clusterhead or a neighboring mote. For this project, the students are required to know the necessary disciplinary and interdisciplinary knowledge in wireless networking and embedded programming. We should add that juniors at Drexel had a course in probability theory at their pre-junior year.

Secondly, this experiment could also be used as a demo to illustrate the concepts in various courses such as Digital Signal Processing, Fundamentals of Communication Theory, Fundamentals of non-deterministic Signal Processing at the undergraduate level, and Statistical Pattern Recognition, Detection and Estimation Theory at the graduate level, where the following theoretical concepts can be illustrated by this experiment.

The students are required to be familiar with embedded programming for Mica2 motes. Other than that, the concepts taught in this experiment are basic and do not require advanced knowledge on classification or detection/estimation theory. The aim of this experiment is to make the students familiar with those concepts within the context of the experiment. A 20-30 minutes lecture at the beginning of this class revisits the concepts of binary hypothesis testing, concepts of false alarm and misdetection associated with a given classifier, the concept of ROC curve within the context of target detection. This experiment is very helpful to students in various aspects. It helps them not only advance their embedded programming skills but also see a real life application of WSN and broaden their view of this subject field. In addition, the students learn some concepts of statistical pattern recognition; basic concepts of target detection system, designing a basic classifier, how to form an ROC curve, and how choosing a different classifier affects the classification. These abstract concepts are most of the time very difficult for students to grasp. However, this real life application helps them understand the material and be able to link the theory to real life problems.

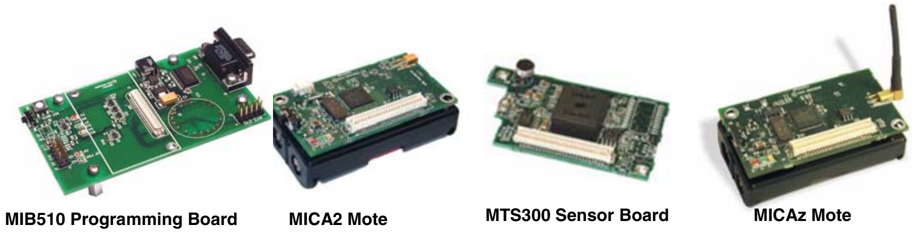


Fig. 1. Hardware used for the experiments (Crossbow Technology Inc. [12])

Experiment Details

This experiment shows students how to build a basic target detection system using WSN as a platform. The used platform and software are TinyOS and NesC language [1]. The hardware components used are given in Figure 1. A light source is used as a target and the light sensor on the sensor board (MTS-300) mounted on top of the Mica2 mote is used as the detecting unit. This simple sensor network application serves as a target detection system where a target shows in the vicinity of a cluster of WSN nodes at various distances from it. The goal here is to convey to the sink based on the "hottest" node in the cluster, whether the cluster decides there is a target (alternative hypothesis) or there is none (null hypothesis) at various false alarm rates.

The experiment/project has three steps. First the students are required to create a sensor network application for data aggregation from a number of nodes. Secondly, as the main step of the application, the students are asked to modify the previous application to build the target detection system. Finally, the students are required to transmit the aggregated data and decision to a PC through Base Station, and show the received data on a PC screen.

The first step of the experiment is data aggregation from three nodes: For example, let's consider three nodes with IDs 3, 4, and 7, where the node with ID 3 is assigned as collector. The detailed steps for data aggregation application are as follows:

- ✧ Program the two nodes (node 4 and 7) such that they collect readings from the light sensors (L4 and L7) every 500msecs and send the readings to the collector node (node 3).
- ✧ Program the collector node (node 3) so that it does the following:
  - ✧ Collect readings from the light sensor (L3) every 500msecs.
  - ✧ Receives all packets sent from other two nodes and keeps track of last light value sent from each unique node including itself (L3, L4, and L7).
  - ✧ Whenever a packet is received,
    - ✧ calculate maximum(L3,L4,L7) and store it as MAXdata,
    - ✧ calculate minimum(L3,L4,L7) and store it as MINdata,
    - ✧ calculate average(L3,L4,L7) and store it as AVGdata,
    - ✧ store the nodeID having the maximum sensor reading as maxID
    - ✧ store the nodeID having the minimum sensor reading as minID
  - ✧ Displays maxID on the LEDs.

In order to test this first step of the experiment, once the students write the programs and install the programs on the motes, the students are required to turn all the nodes and observe the output on the LEDs by changing the location of the light source. The LEDs on node 3 should show the ID of the node having the maximum light sensor reading.

The second and the main step of the experiment is target detection. The students are required to modify the program for the collector node in the first step. The detailed steps for the programming are given in the following:

- For the first 40 seconds, collect readings from the light sensor every 500msecs and store it in a buffer.
- After collecting first set of readings (when there is a target in the vicinity), turn on red LED for 3 sec.
- For the next 40 seconds collect readings from the light sensor every 500msecs and store it in a buffer.
- After collecting second set of readings (when there is no target in the vicinity) turn on green LED for 3 sec.
- Calculate Receiver Operating Characteristics (ROC) curve based on these two sets of readings:
    - Create histograms (empirical distributions) for both set of readings.
    - Calculate points on the ROC curve (probability of detection versus false alarm curve) from the histograms.
    - Store the threshold values corresponding to each point on the ROC curve.
    - Calculate the area under the ROC curve (Az) and store it as Az.
- Receive all packets sent from other two nodes and calculate the maximum sensor reading (MAXdata) and the ID of the node that has the maximum sensor reading (maxID).
- Select the corresponding threshold value from the ROC for a false alarm rate of 0.1 (α=0.1).
- Compare the maximum sensor reading (MAXdata) to the threshold (THR).
    - If MAXdata>THR then turn on green LED for 1 second (target is detected) and then display maxID on the LEDs for 1 second. And, set the variable TARGET=1.
    - Else, turn on red LED (no target) for 2 seconds, and set the variable TARGET=0.

The students are provided with a sample algorithm to create an ROC curve for any two data arrays. In order to test this main step of the experiment, the students should write the program and install the program on the collector node (node 3), and keep the other nodes with the same program from the previous step. They are required to draw two concentric circles having diameters ~4 and ~8 inches on a paper. Then they will put the node with ID 3, on the center of the circles (the light sensor should be at the center) and turn on node 3. This node is used to obtain the Receiver Operating Characteristics (ROC) curve using some light sensor readings. They will hold the light source so that it shines its light in between the circles (2"<R<4") and change the location of the light source staying in that region for 40 seconds (until they see the red LED blink) trying to cover all of that area in the given time. Then they will hold the light source so that it shines its light inside the inner circle (R<2") and change the location of the light source staying in that region for 40 seconds (until they see the green LED blink) trying to cover all of that area in the given time. Afterwards, the program computes the ROC curve and calculates the corresponding threshold for the selected false alarm rate (e.g., α=0.1) and gets ready for target detection.

Once the data gathering for forming the classifier is completed, they will turn on the other nodes with IDs 4 and 7, and change the location of the light source and view the LEDs on node 3. A target is assumed to be detected when it is within distance<2". In that case green light on node 3 blinks for 1 second, and the node ID that detects the target is shown on the LEDs for the next 1 second. If the target is outside the range >2", then no target is detected. In that case, red light on node 3 blinks for 2 seconds. During the whole experiment the light source should be held at the same height. The experiments could also be repeated for circles of different sizes; the size of the circle will change the resolution of the detection system. Also the students are asked to try different false alarm values (α in the program) keeping the same circle and observe the change in detection (0<α<1). The students can observe how the design of a classifier depends on training, and selecting some parameters.

The final step of the experiment is transmitting the aggregated data and decision to a PC through a Base Station: The node with ID 3 aggregates data from the other nodes (ID 4 and 7) and itself, and makes required calculations and sends the aggregated data and decision to Base Station (ID 1). And, the received data is shown on a PC screen. The detailed steps for the programming follow:

✧ Install BaseStation application to the node with ID 1 (this application is provided by TinyOS [1]).

✧ Add the required lines to your header file and makefile to create and display a message structure including the variables MAXdata, MAXnodeid, AVGdata, MINdata, MINnodeid, TARGET, Az, ALPHA, and THR.

In order to test this final step of the experiment, the students type the suitable command Cygwin [14] screen to see for their own defined messages. They observe the output on screen by changing the location of the light source and verify the detection results shown on LEDs by comparing them to the output on the screen.

The figure for Experiment 1 is shown in Figure 2. This experiment can be modified according to the allocated time in the course.

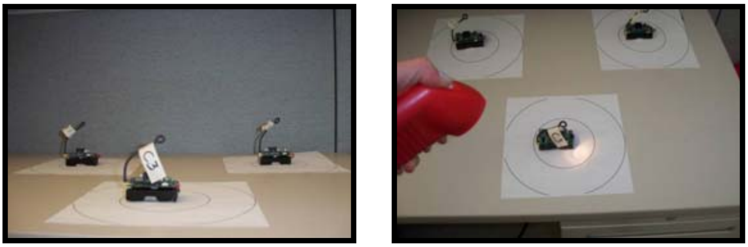


Fig. 2. Pictures showing the motes for Experiment 1

2.2 Experiment 2 – Local, Selected and Consensus Decision on Target Detection

Where Would the Experiment Fit?

It is natural to import this experiment (or a variation of it) to courses on Digital Signal Processing, Fundamentals of Communication Theory, Fundamentals of non-deterministic Signal Processing at the undergraduate level, and Statistical Pattern

Recognition, Detection and Estimation Theory at the graduate level. A demo of the experiment would be helpful for students to improve on their understanding of some of the fundamental concepts in these courses.

Experiment Details

This experiment shows students how to build an advanced target detection system using wireless sensor networks as a platform. The used platform and software are TinyOS, NesC language [1] and Matlab [13]. The hardware components used are given in Figure 1. A light source is used as a target and the light sensor on the sensor board (MTS-300) mounted on top of the Mica2 mote is used as the detecting unit as in experiment 1 (Section II-A).

An experiment is developed where a target shows in the vicinity of a cluster of nodes at various distances from it. The goal here is to convey to the sink either based on the "hottest" node in the cluster, on a select few (concept of maximum redundancy) or on all the nodes in the cluster whether the cluster through its cluster head decides there is a target (alternative hypothesis) or there is none (null hypothesis) at various false alarm rates. The students should know how to use radio communication between nodes, messaging and displaying data and performing simple signal processing at the node level.

The experiment has three main steps. First the students are required to collect data to the collector node and send the aggregated data to MATLAB through a PC. Secondly, the students are asked to determine the histograms of the node sensor output as a function of the distance of the target from the node and to obtain the ROC curves (probability of detection versus false alarm curve) from these histograms. Finally, they are required to select appropriate thresholds from the ROC curves and test the 3 various protocols in detecting the target.

The first step of the experiment is gathering the required data to determine the empirical distribution (histograms). Let's consider four nodes with IDs 1, 3, 4, and 7, where the node with ID 1 is assigned as the collector node. The data aggregation program in Experiment 1 will be edited to include four nodes and additionally the light sensor readings for each node in the cluster will be stored with the detailed steps as given in the following.

- Program the three nodes (node 3, 4 and 7) such that they collect readings from the light sensors (L3, L4 and L7) every 500msecs and send the readings to the collector node (node 1).
- Program the collector node (node 1) so that it does the following:
    - Collect readings from the light sensor (L1) every 500msecs.
    - Receives all packets sent from other three nodes and keeps track of last light value sent from each unique node including itself (L1, L3, L4, and L7).
    - Whenever a packet is received,
        - calculate maximum(L1, L3,L4,L7) and store it as MAXdata,
        - calculate minimum(L1, L3,L4,L7) and store it as MINdata,
        - calculate average(L1, L3,L4,L7) and store it as AVGdata,
        - store the nodeID having the maximum sensor reading as maxID
        - store the nodeID having the minimum sensor reading as minID
        - store the L1, L3, L4, and L7 and corresponding node ID's
    - Displays maxID on the LEDs.

The students are asked to send the collected data packet to a PC in a similar fashion as in the last step of Experiment 1. They are required to read the collected packet in MATLAB, and may create a Graphical User Interface (GUI).

The second step of the experiment is to determine the histograms of the nodes' sensor outputs as a function of the distance of the target from the nodes, where a target is assumed present if it is within a radius R of a node and assumed non-present if the distance exceeds the threshold distance R. Again, similar distance values can be used as in experiment 1 (Section II-A) where the target is assumed to be present if it is within a distance R<2" to a node. From these set of histograms, ROC curves (probability of detection versus false alarm curves) are determined. The students are asked to continue the experiment using MATLAB. The MATLAB program should do the following.

- ✧ For a specified time period (e.g. 60 seconds) store the readings from all 4 nodes as well as AVGdata, and average of maximum 3 readings from nodes {MAX3data = [(3*AVGdata-MINdata)/4]} while the light source shines light inside the inner circle for the whole cluster. Create the empirical distribution (histograms) for each of the 4 nodes as well as AVGdata, and MAX3data.
- ✧ For the next time period, store the readings from all 4 nodes as well as AVGdata, and MAX3data while the light source shines light in between the circles for the whole cluster. Create the empirical distribution (histograms) for each of the 4 nodes as well as AVGdata, and MAX3data.
- ✧ Using the histograms for both set of readings, calculate ROC curves for each of the nodes, AVGdata and MAX3data.
  - ✧ Create histograms (empirical distributions) for both set of readings.
  - ✧ Calculate points on the ROC curve from the histograms.
  - ✧ Store the threshold values corresponding to each point on the ROC curve.
  - ✧ Calculate the area under the ROC curve (Az).

In order to test this main step, the students are required to specify the locations of nodes in a given cluster, and for those fixed locations, they are required to draw the circles as in Experiment 1 for all the four nodes. The union of those circles will give the overall target detection range for the whole cluster. Figure 3 shows a sample cluster of four nodes and target detection range, the red curve shows the union of inner circles and the black curve shows the union of outer circles. They will hold the light source so that it shines its light inside the overall inner circle for a specified time period and create the empirical histograms. Then they will hold the light source so

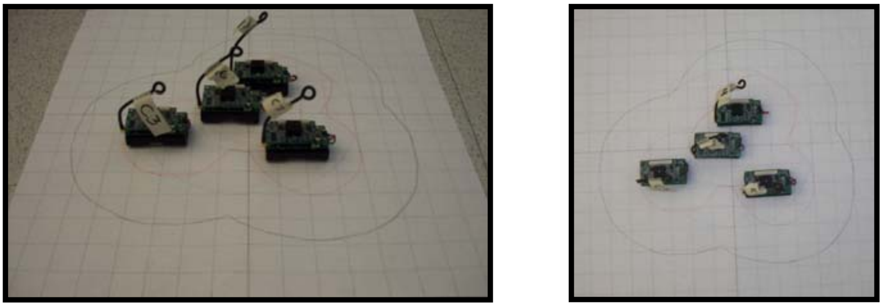


Fig. 3. Pictures showing the motes for Experiment 2-3 for one cluster

that it shines its light between the circles for the whole cluster for the next specified period of time and create the empirical histograms. Afterwards, the program computes the ROC curves and calculates the corresponding thresholds for various false alarm rates. The GUI for Experiment 2 Demo is shown in Figure 4. The ROC curves for each case are also shown in the figure.

Finally, from the ROC curves the appropriate thresholds are extracted against which the measured sensed data or some statistics of it will be compared to for accepting or rejecting that there is a target. The detailed steps for the programming are given in the following:

- ✧ Give the user the option of selecting the decision criteria. For each of the decision criteria (decision criteria 1: hottest node, decision criteria 2: select a few (max. red.), decision criteria 3: all the nodes) write a program which
  - ✧ Selects the corresponding threshold values from the corresponding ROC for a given false alarm rate (e.g. α=0.1).
  - ✧ Compares the corresponding measured sensed data for the selected protocol to the threshold.
    - ✧ Case1: Compares sensed MAXdata to the threshold extracted from the ROC curve of the node with node ID = maxID.
    - ✧ Case2: Compares sensed MAX3data to the threshold extracted from the ROC curve of MAX3data.
    - ✧ Case3: Compares sensed AVGdata to the threshold extracted from the ROC curve of AVGdata.
  - ✧ If >THR then target is detected in the vicinity of the cluster, else target is not detected in the vicinity of the cluster.
  - ✧ Display the results on MATLAB screen.

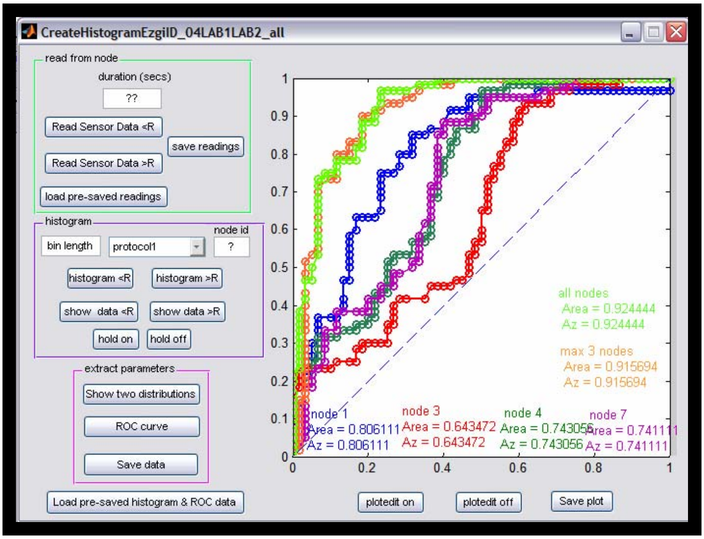
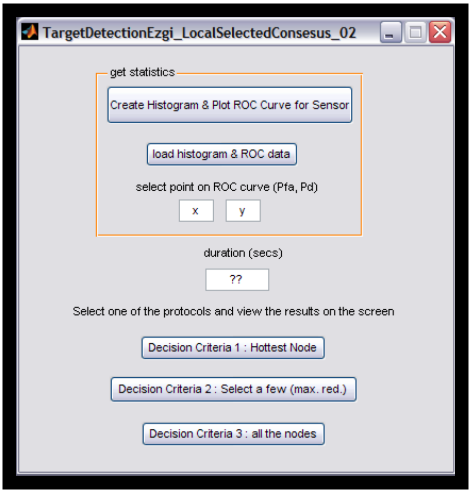


Fig. 4. GUI for Experiment 2 Demo

In order to test this step, the students will place the target at a fixed position then turn all the nodes in the cluster and test the 3 various protocols in detecting the target. The students will move the target closer to the cluster as well as away from it and check their results against the ground truth (no target is in the cluster vicinity if the target is R distance away from the closest node in the cluster). They will also check on how turning off one of the sensor nodes (simulating sensor node damage) can impart the decision on each of the 3 protocols. The GUI-2 for Experiment 2 Demo is shown in Figure 5.


TargetDetectionEzgi_LocalSelectedConsesus_02
get statistics
Create Histogram & Plot ROC Curve for Sensor
load histogram & ROC data
select point on ROC curve (Pfa, Pd)
x
y
duration (secs)
??
Decision Criteria 1 : Hottest Node
Decision Criteria 2 : Select a few (max. red.)
Decision Criteria 3 : all the nodes

Select one of the protocols and view the results on the screen

Fig. 5. GUI-2 for Experiment 2 Demo

2.3 Experiment 3 – Motion Tracking

Where Would the Experiment Fit?

A demo of the experiment would be very helpful for students to improve on their understanding of some fundamental concepts in courses such as Fundamentals of Intelligent Decision, and Fundamentals of Imaging at the undergraduate level, and Statistical Pattern Recognition, Detection and Estimation Theory, Computer Vision at the graduate level.

The students are expected to be familiar with all the prerequisites stated for experiment 2 (Section II-B.2) such as basic probability and statistics concepts related to simple binary hypothesis testing. And the students should be comfortable with programming the nodes for radio communication between nodes, messaging and displaying data and doing simple signal processing at the node level. In addition to the requirements given for the previous experiment, knowledge of simple motion equations and line fitting is needed.

Experiment Details

This experiment shows students how to build a motion tracking system using wireless sensor networks as a platform. The used platform and software are TinyOS, NesC language [1] and Matlab [13]. The hardware components used are given in Figure 1. The target and the detecting unit are the same as in the previous experiments.

This experiment builds on the previous one (Section II-B) by having the target now moving amongst a few clusters of sensors (3 or 4 clusters that are well separated distances greater than the minimum R distance for target identification). The students will experiment with what they have developed in the previous experiment for target detection for motion tracking.

The experiment has three main steps. First the students are asked to track the target as it is moving amongst the nodes and clusters, and record the location of the target and the time. Secondly, the students are required to fit a straight line through the recorded points, and based on a constant velocity assumption they are asked to compute the velocity of the target. Finally, they are required to determine whether the target will be sensed by a given cluster in the path of the moving target and predict the time when the target should reach a given cluster. Additionally, the students will be asked to determine the rough location of a new cluster of which the location is unknown based on the pre-computed velocity up to that point.

The students are asked to add on to the MATLAB code they developed for Experiment 2. First, the students are asked to edit the program as given in the following:

✧ For each of the decision criteria, if a target is assumed to be detected in the vicinity of cluster n1 at a given time t, determine the hottest node in the cluster and record that point for that time as the closest point for the trace of the target motion and keep a track of the cluster id.
✧ At time t + Δt, when a new hottest node is found record that new point as the next point (new target location).
✧ Continue tracking the target as long as the target is within the R limit of cluster n1.
✧ When the target becomes invisible (no target in the vicinity of cluster n1) and enters the R space of another cluster (e.g. cluster n2) after MΔt time interval, record new points including the new cluster id.
✧ At each point display the time and location of the target on MATLAB screen as well as the cluster id.
✧ Plot the path of the target on computer screen.

The students will move the target slowly amongst clusters and validate the results on computer screen. Once this first step of the experiment is validated to be working properly, the following steps will be added to the program as given in detail in the following:

✧ At a given time (tn) fit a straight line through the recorded points up to that time (p1 through pn). Give the user the option to select the order of the polynomial fit.
✧ Based on a constant velocity assumption, compute the velocity of the target. If the first recorded time is t1, then at a given time tn, the velocity is computed as: [(length of the fitted line from p1 to pn) / (tn-t1)]
✧ Plot the fitted line and display the computed velocity on computer screen.

The students will observe the results on computer screen as the target moves amongst the clusters, and once validated they will move on to the final step of the experiment and edit the code as given in the following:

- ✧ Using the velocity and fitted line predict the time when the target should reach a given cluster in the path of the moving target.
- ✧ Determine whether or not it will be sensed based on that R vicinity space threshold and by which node.
- ✧ Display the decisions on MATLAB screen.

Optionally, the experiment could be developed further by adding a new cluster to the scene whose location is unknown to the program. Then, the new task would be determining the rough location of that new cluster once that cluster senses the target based on the estimated velocity and path of the target up to that point.

The GUI for Experiment 3 Demo is shown in Figure 6.

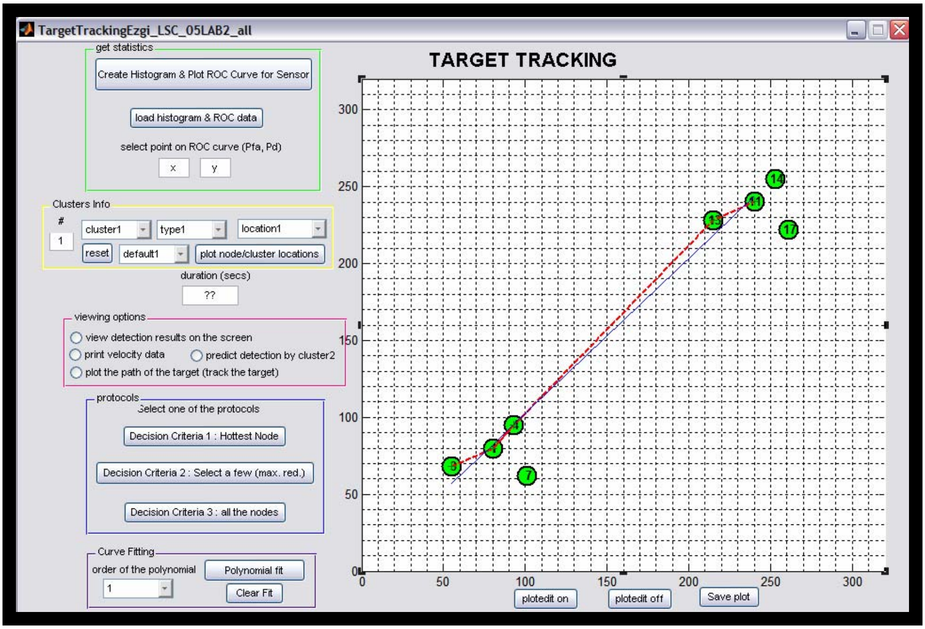

TargetTrackingEzgi_LSC_05LAB2_all
get statistics
Create Histogram & Plot ROC Curve for Sensor
load histogram & ROC data
select point on ROC curve (Pfa, Pd)
x
y
Clusters Info
#
1
cluster1
type1
location1
reset
default1
plot node/cluster locations
duration (secs)
??
viewing options
○ view detection results on the screen
○ print velocity data
○ predict detection by cluster2
○ plot the path of the target (track the target)
protocols
Select one of the protocols
Decision Criteria 1 : Hottest Node
Decision Criteria 2 : Select a few (max. red.)
Decision Criteria 3 : all the nodes
Curve Fitting
order of the polynomial
1
Polynomial fit
Clear Fit
TARGET TRACKING
300
250
200
150
100
50
0
0
50
100
150
200
250
300
14
11
13
17
4
4
8
7
plotedit on
plotedit off
Save plot

Fig. 6. GUI for Experiment 3 Demo

3 Assessment Survey and Evaluation

3.1 Assessment Survey

The evaluation of the lab activities and in-class demos is done with the help of an assessment survey questionnaire judiciously designed to reflect the degree of enhanced learning using the WSN hands-on experiments platform. The responses and feedback from students are not only to evaluate the performance of the proposed experiments for adding to students' learning process, but they are also useful as feedback for future fine tuning of these modular experiments as pedagogical learning tools.

The questions asked on the designed assessment survey are:

1. How much did you know about Bayes classifier, Neyman Pearson (NP) classifier and likelihood ratio tests prior to being exposed to this experiment/demo?
2. How much did you know about Receiver Operating Characteristics (ROC) and the use of the area under the ROC for measuring the “effectiveness” of a classifier?
3. How much does the tracking experiment with the sensor nodes help in seeing the practical side and applicability of these theoretical concepts by introducing them within a very practical setting of detecting a target using sensor network?
4. How much would you like to see more hands-on demos that illustrate and link the theory to real applications?
5. After seeing the demo, to what extent had the demo raised interest in you in knowing more about the field of sensor network?
6. Students were also asked to write their own comments if they wish to.

The answer choices ranging from a score of 1 to a score of 5 were: 1: none, 2: very little, 3: average, 4: good, 5: superb.

3.2 Results of the Survey

The experiments are demonstrated in three different courses, ECE-C490: Wireless Embedded Networks Lab (junior/senior undergraduate level), ECE-S435: Digital Signal Processing II (DSP) (senior undergraduate level), ECE-S690: Statistical Pattern Recognition (SPR) (graduate level). In addition to these demonstrations, experiment 1 is given as a final project / final lab assignment for the course ECE-C490: Wireless Embedded Networks Lab.

The statistics (mean, standard deviation) of the scores for each question for each course are given in Table 1. The histograms of the student scores are also given in Figure 7.

Table 1. Statistics of Student Answer Scores

		Q1	**Q2**	**Q3**	**Q4**	**Q5**
Wireless Embedded Networks Lab	mean		**1.8**	**3.8**	**4.1**	**4.3**
	standard deviation		1.2	0.4	0.6	0.7
DSP II	mean	**2.8**	**2.7**	**3.8**	**3.8**	**3.9**
	standard deviation	0.9	0.9	1.1	0.8	0.9
SPR	mean	**2.8**	**2.9**	**3.9**	**3.8**	**3.2**
	standard deviation	0.7	1.1	0.8	1.0	1.1
overall	mean	**2.8**	**2.5**	**3.8**	**3.9**	**3.8**
	standard deviation	0.8	1.1	0.8	0.8	1.0

The students were asked to write their comments if they wished to in the 6th part of the questionnaire. The comments from students are given below for each or the courses:

Comments from the class ECE-C490: Wireless Embedded Networks Lab

“Was a really fun class…”, “Would definitely like to know more about how the ROC etc works…”, “Really interesting lab, very informative…”, “I would like to see more things like this being taught in this class…”, “This was an excellent demo and application and was very helpful…”

Comments from the class ECE-S690: Statistical Pattern Recognition

“Demos and hands-on homework/projects are always way more effective in relaying the concepts at hand. Using these types of things keep me more interested…”, “More demonstrations and experiments would make the material, which is largely abstract…”, “I really enjoyed seeing real world applications of the material…”

Comments from the class ECE-S435: Digital Signal Processing II (Statistical Signal Processing)

“I learned a lot, both from the lecture and the demo. Both very applicable…”, “Really nice demos!”

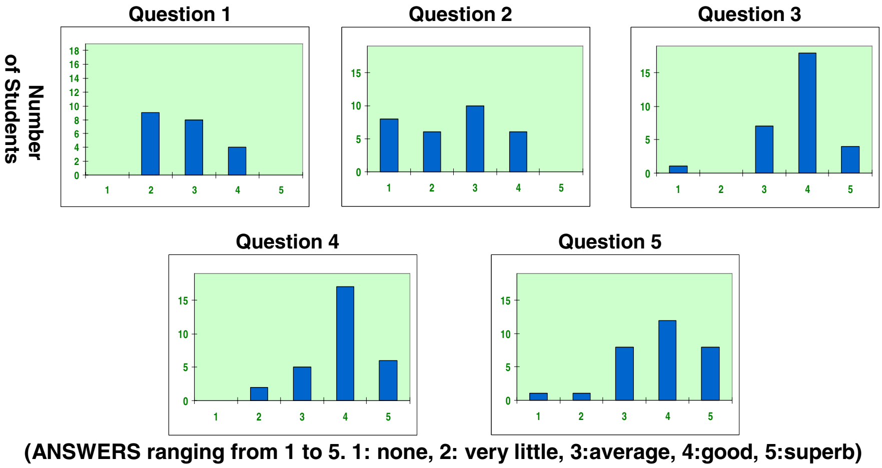


Fig. 7. Histograms of Student Answer Scores

3.3 Conclusive Assessments

Based on the survey results (Table 1, Figure 7, and comments from students) the assessments reached are:

- ✧Students entered the Digital Signal Processing (DSP) and Statistical Pattern Recognition (SPR) courses with little to average knowledge of the Bayes classifier, NP classifier and likelihood ratio tests.
- ✧Students entered the DSP and SPR courses with little to average knowledge of the Receiver Operating Characteristics (ROC) and the use of the area under the ROC for measuring the “effectiveness” of a classifier.
- ✧Senior DSP and Graduate SPR students found that the tracking experiment with sensor nodes enhanced their understanding of the underlying theoretical concepts.

- ✧Undergraduate and graduate students want more hands-on demos which link theory to real applications.
- ✧The target detection in sensor network approach motivated students to further learn about the field of sensor network.
- ✧Undergraduates in the Embedded Network Lab course entered with no or little knowledge of Target Detection in Sensor Networks.
- ✧Undergraduates in the Embedded Network Lab course found that the tracking experiment with sensor nodes enhanced their understanding of the underlying theoretical concepts.
- ✧Undergraduates in the Embedded Network Lab course want more hands-on demos which link theory to real applications as well as hands-on projects built around the demos and more time to engage in the lab, possibly a 2-week lab instead of 1-week.
- ✧Undergraduates in the course dealing with the field of sensor networks wanted more demos that use the sensor network as an application platform as well as wanted a formal course on sensor network.

4 Conclusions

Through hands on experiments built on MICA2 motes (with on-board sensors, wireless communication antennas, and its own processors programmed using NesC), we showed how various theoretical abstract concepts in a variety of ECE courses can be made tangible and easier to understand. This is particularly true for those theoretical concepts that are traditionally hard to grasp and comprehend using the traditional lecturing/homework/exams method. The proposed experiments based on Wireless Sensor Networking platform are shown to be exciting pedagogical learning tools and portable to various multi-disciplinary existing courses ranging from junior/senior to graduate level.

This work was presented at Drexel Research Day 2008 (where demos and posters are exhibited and presented and viewed by the University at large as well as by visitors from outside the University and also from the industry) and the significance of this work was awarded with “Best Poster Presentation Award” under the category of “Innovation in Education and Outreach.”

A variation of the proposed experiments was also used in the NSF high school teacher education program offered at Drexel University (Research Experiences for Teachers (RET)) and is used in a local high school lab on technology and physics. The emphasis of that experiment was to illustrate law of motion and dynamics through a WSN on tracking and velocity and acceleration updates.

In summary, we successfully met our educational goals through designed WSN experiments. The developed experiments were modular so and were easily imported into other courses outside of the sensor networks area (the platform used for the experiments), such as, for instance, in a course on detection and estimation or on stochastic processes, which are traditionally very dry and could be enriched with applications that would clarify the concepts taught and motivate the students to learn and improved their understanding of various traditionally abstract concepts in a wide range of courses in Electrical and Computer Engineering. This type of excitement helps to attract, excite, and retain students in electrical engineering by exposing them to real life applications beginning in the early stages of the curriculum. The abstract and difficult to grasp theory is very well presented in the proposed experiments and improved the understanding of the students on various concepts in a wide range of courses in Electrical and Computer Engineering.

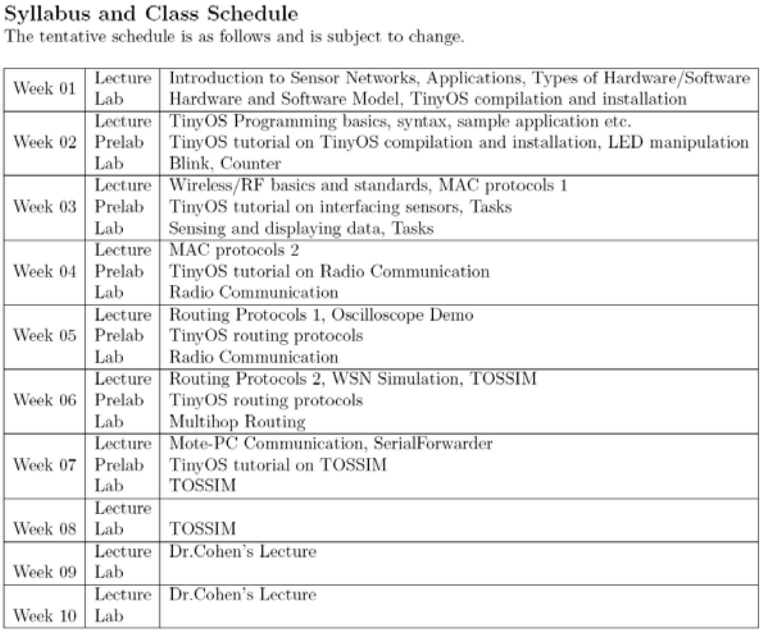
Acknowledgments. We would like to thank Dr. Jaudelice Cavalcante de Oliveira for her valuable ideas/comments while designing Experiment 1. This work is supported by the National Science Foundation, Division of Undergraduate Education, under grant # 0633576.

References

1. TinyOS, http://www.tinyos.net
2. The "moteLab" at Harvard University, http://motelab.eecs.harvard.edu/
3. The Wireless Sensor Network Laboratory (WSNL) at Stanford University, http://wsnl.stanford.edu/
4. The Embedded Networks and Applications Laboratory "ENALAB" at Yale University, http://www.eng.yale.edu/enalab/
5. Networked & Embedded Systems Laboratory (NESL) at University of California, Los Angeles, http://nesl.ee.ucla.edu/
6. UCLA's Center for Embedded Networked Sensing, http://cens.ucla.edu/
7. The MANTIS (MultimodAl NeTworks of In-situ Sensors) project lab, a part of the Computer Science Department at the University of Colorado, Boulder, http://mantis.cs.colorado.edu/tikiwiki/tikiindex.php
8. Abrach, H., Bhatti, S., Carlson, J., Dai, H., Rose, J., Sheth, A., Shucker, B., Deng, J., Han, R.: MANTIS:System Support For MultimodAl NeTworks of In-situ Sensors. In: 2nd ACM International Workshop on Wireless Sensor Networks and Applications, WSNA, pp. 50–59 (2003)
9. Bhatti, S., Carlson, J., Dai, H., Deng, J., Rose, J., Sheth, A., Shucker, B., Gruenwald, C., Torgerson, A., Han, R.: MANTIS OS: An embedded multithreaded operating system for wireless micro sensor platforms. Mobile Networks & Applications 10, 563–579 (2005)
10. Mache, J., Bulusu, N., Tyman, D.: Making Sensor Networks Accessible to Undergraduates Through Activity-Based Laboratory Materials. In: Proceedings of IEEE SECON (2008)
11. Mache, J., Bulusu, N., Tyman, D.: Sensor Network Lab Exercises Using Java and Sun Spots. Poster at ACM SIGCSE (2008)
12. Crossbow Technology Inc., http://www.xbow.com
13. Matlab, http://www.mathworks.com/
14. Cygwin, http://www.cygwin.com

Appendix

ECE-C490 Wireless Embedded Networks Lab[1]



Syllabus and Class Schedule

The tentative schedule is as follows and is subject to change.

Week 01	Lecture	Introduction to Sensor Networks, Applications, Types of Hardware/Software
	Lab	Hardware and Software Model, TinyOS compilation and installation
Week 02	Lecture	TinyOS Programming basics, syntax, sample application etc.
	Prelab	TinyOS tutorial on TinyOS compilation and installation, LED manipulation
	Lab	Blink, Counter
Week 03	Lecture	Wireless/RF basics and standards, MAC protocols 1
	Prelab	TinyOS tutorial on interfacing sensors, Tasks
	Lab	Sensing and displaying data, Tasks
Week 04	Lecture	MAC protocols 2
	Prelab	TinyOS tutorial on Radio Communication
	Lab	Radio Communication
Week 05	Lecture	Routing Protocols 1, Oscilloscope Demo
	Prelab	TinyOS routing protocols
	Lab	Radio Communication
Week 06	Lecture	Routing Protocols 2, WSN Simulation, TOSSIM
	Prelab	TinyOS routing protocols
	Lab	Multihop Routing
Week 07	Lecture	Mote-PC Communication, SerialForwarder
	Prelab	TinyOS tutorial on TOSSIM
	Lab	TOSSIM
Week 08	Lecture	
	Lab	TOSSIM
Week 09	Lecture	Dr.Cohen's Lecture
	Lab	
Week 10	Lecture	Dr.Cohen's Lecture
	Lab	

[1] This course is developed by Dr. Jaudelice Cavalcante de Oliveira and was offered during the 2007-2008 Winter Quarter in the Department of Electrical and Computer Engineering at Drexel University.