International Electronic Conference on Sensors and Applications

1-16 June 2014

Localization with RSSI values for Wireless Sensor Networks: An Artificial Neural Network Approach

By: **Shiu Kumar (Department of Electronics Engineering, Mokpo National University, Korea)**

Summary Layout

- **Introduction - problem definition**
- **Data Collection**
- **Experimental Setup**
- **Training the ANN**
- **Proposed ANN** - implementation
- **Results and Discussion**
- **Conclusions**

Introduction

- WSNs have broad applications in scientific data gathering, performing search and rescue operations, real-time information processing for disaster response, monitoring and surveillance, security, and military applications.

- One of the fundamental challenges and active research areas in wireless sensor networks is node localization.

- Node localization refers to determining the physical location of each node in the network.

- Most WSN applications need to have location information of the sensor nodes in order to make the measured data significant.

Introduction

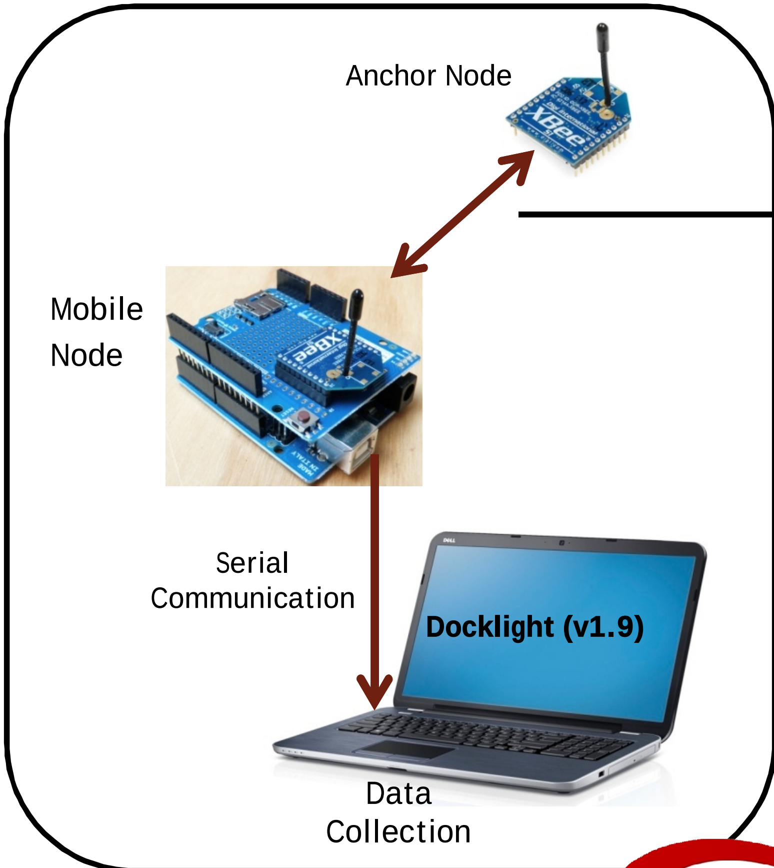
- It is impractical to note down or record the location of each of the sensor nodes during the time of deployment as WSNs typically consists of a large number of spatially distributed sensor nodes.

- Node localization is required to:
  - report the origin of events
  - assist group querying of sensors
  - routing
  - and to answer questions about the network coverage

- Location information is used in many location-aware applications such as navigation, tracking, searching, and rescue operations.

International Electronic Conference on Sensors and Applications
1-16 June 2014

Data Collection

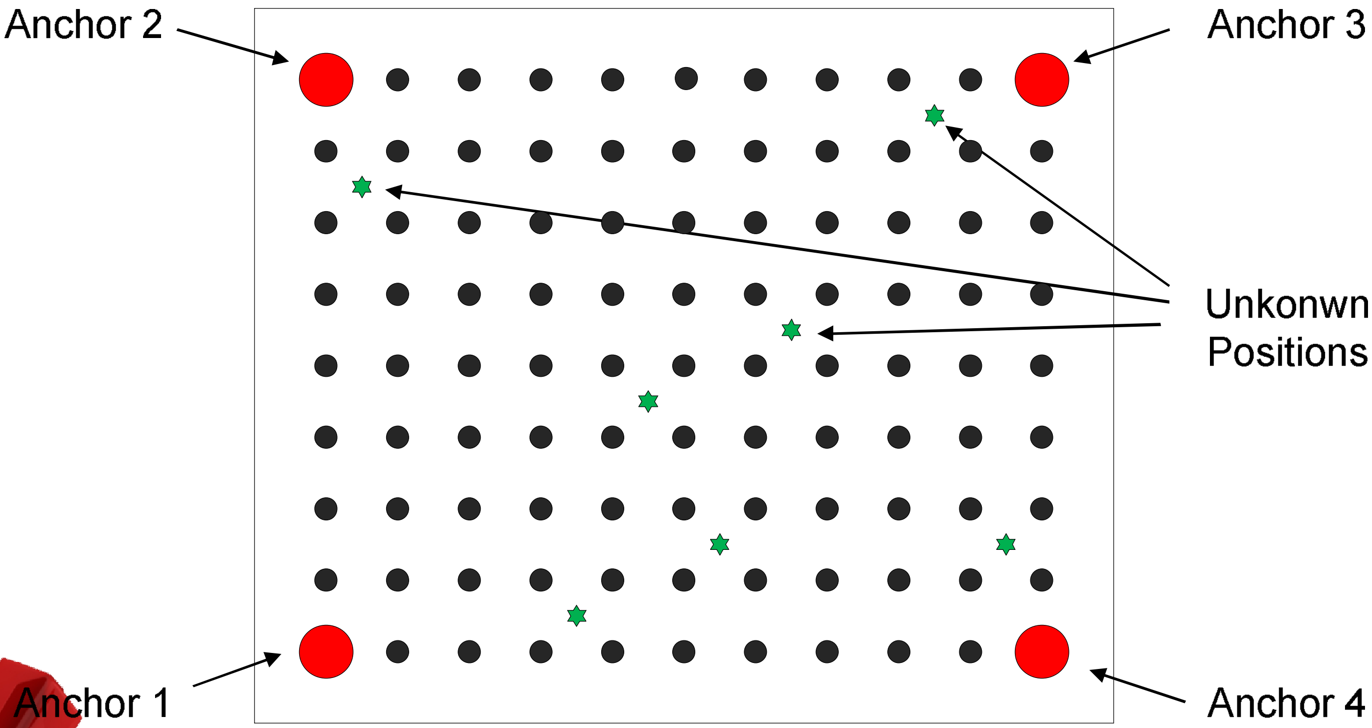


- Sensor node → XBee Series 2
  → Arduino Uno
- Location determination → using Received Signal Strength Indicator (RSSI) value.
- Command XBee explicitly using DB command → returns RSSI value.

Artificial Intelligence

Experimental Setup Layout

International Electronic Conference on Sensors and Applications
1-16 June 2014



Red → Anchor nodes
Black → Training data points at 0.50m [(9 x 11) – 4 = 95]
Green → Testing data points (unknown positions)
Environment → Lab → containing tables, chairs, computers

Proposed Method: Training

International Electronic Conference on Sensors and Applications
1-16 June 2014

- Multi-Layer Perceptron (feed-forward) Neural Network.
  - Software → Matlab
  - Learning Algorithms Evaluated → Levenberg-Marquardt (LM), Bayesian Regularization (BR), Resilient Back-propagation (RP), Scaled Conjugate Gradient (SCG) and Gradient Descent (GD).
- Data Set : 25 x 95 = 2375 data set for known positions (60% Training, 20% Validation, 20% Testing)
  : 7 x 15 = 105 data set for unknown positions
- Dataset structure

$$data = \begin{bmatrix} R_{11} & R_{12} & R_{13} & R_{14} & X_1 & Y_1 \\ R_{21} & R_{22} & R_{23} & R_{24} & X_2 & Y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ R_{m1} & R_{m2} & R_{m3} & R_{m4} & X_m & Y_m \end{bmatrix}$$

Proposed Method: Training

- Dataset Structure → R_{ij} denotes the RSSI values of the signal perceived from the j^{th} anchor node, at the i^{th} reference point while X_i and Y_i denote the x and y coordinates of the i^{th} reference point.
- Different Artificial Neural Network (ANN) structures were tested. → keeping in mind the computational complexity, cost and localization accuracy a 12-12-2 structure was selected.
- Using the selected ANN structure, all the learning algorithms were used for training and evaluating their performance.
- The ANN with BR learning algorithm was finally selected (explained in results).

Proposed Method: Training

International Electronic Conference on Sensors and Applications
1-16 June 2014

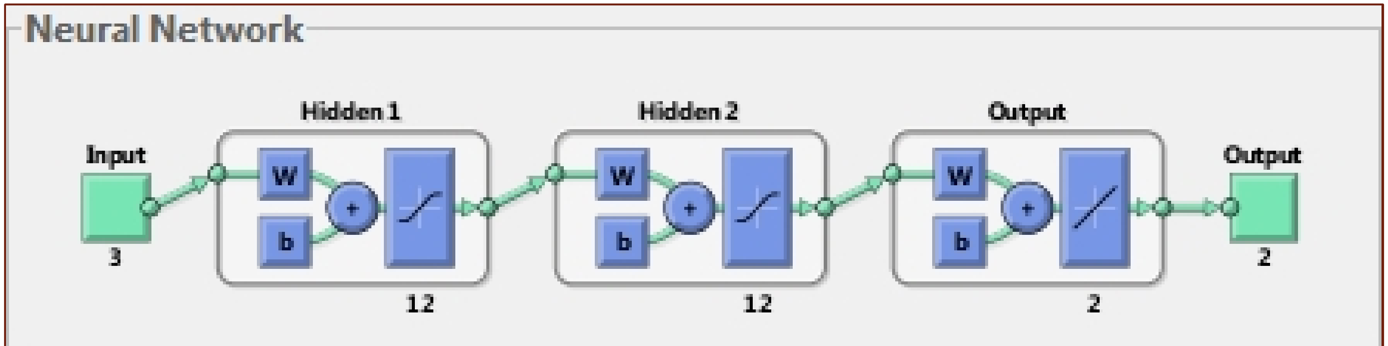
- During training, the best solutions for each type of learning algorithm were selected depending on the validation checks.
- Then the final ANN obtained above were used to evaluate how well they performed on the test data.
- The Error Calculation is done using the following formula:

$$e=\sum_{i=1}^{n}\frac{1}{n}\sqrt{(x_i-x_{oi})^2+(y_i-y_{oi})^2}$$

- Where n is the number of samples, (x_{oi}, y_{oi}) is the actual and (x_i, y_i) is the estimated coordinates of the mobile node at the i^{th} test data set.

International Electronic Conference on Sensors and Applications
1-16 June 2014

Proposed ANN



Algorithms

Data Division:	Random (dividerand)
Training:	Bayesian Regulation (trainbr)
Performance:	Mean Squared Error (mse)
Derivative:	Default (defaultderiv)

- Activation functions: 1st & 2nd layer → hyperbolic tangent sigmoid
  : 3rd layer → pure line
- Inputs: RSSI value from the 4 anchor nodes
- Outputs: *x* and *y* coordinates of the mobile node

Implementation

- The parameters obtained from the trained ANN were used to implement the ANN on the Arduino platform using the equation given below:

$$\begin{bmatrix} x & y \end{bmatrix} = \tanh\left[\tanh\left(R \bullet \left(W_k^{(1)} \right)^T + b_k^{(1)} \right) \bullet W_k^{(2)} + b_k^{(2)} \right] \bullet W_k^{(3)} + b_k^{(3)}$$

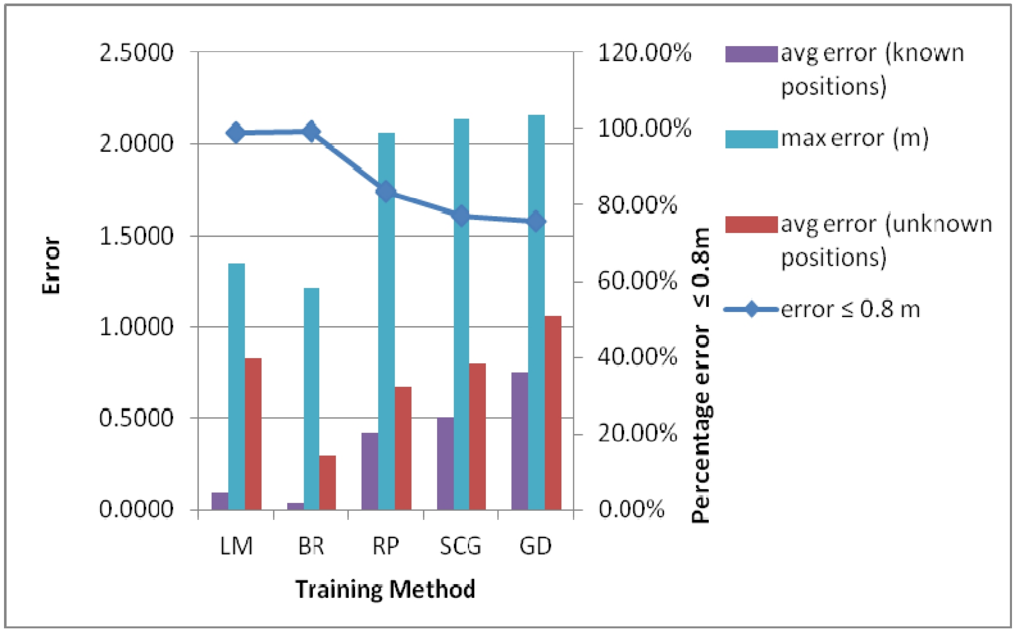
- R → Input of the ANN (RSSI values of signals from anchors) [1x4]
- $W_k^{(l)}$ → Weight vector of k^{th} node of l^{th} layer
- $b_k^{(l)}$ → Bias vector of k^{th} node of l^{th} layer

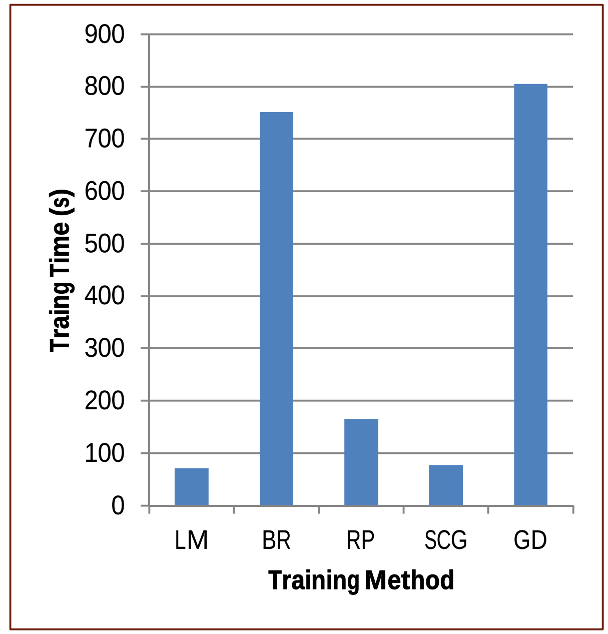
- A learned ANN can be implemented using other programming languages in a similar way.

Artificial Intelligence

International Electronic Conference on Sensors and Applications
1-16 June 2014

Results





Graph showing the average, maximum error and percentage of time error is less than 0.8m for different training algorithms.

Graph showing the time taken to train the ANN for different training algorithms

Results and Discussion

- The ANN learned from BR algorithm was selected as it gave maximum error of 1.21 m, average error of 0.04 m for test at known positions and average error of 0.30 m for test at unknown positions.

- The errors obtained in the ANN learned from BR algorithm were less than that of ANN learned from all other methods evaluated.

- 99 percent of the time the localization error for the selected ANN was less than 0.80 m.

- However, BR algorithm takes about 751 seconds for training the network, which is the 2nd highest.

Artificial Intelligence

International Electronic Conference on Sensors and Applications
1-16 June 2014

Results and Discussion

- Since offline training is carried out only to obtain the ANN parameters for implementation on Arduino platform, the training time was not considered.
- For applications where online training will be performed, the LM learning algorithm is recommended.
- Mamdani & Sugano Fuzzy Inference System (FIS) [10] used 121 anchor nodes and obtained average localization error of 3.0 m.
- A neural network approach in [9] obtained an average localization error of 0.4855 m for 2D movement using 4 anchor nodes.

Results and Discussion

- The MLP neural network is chosen due to its best trade-off between the accuracy and memory requirements among the other types of neural networks.

- The proposed ANN achieved a better localization performance compared to other methods such as [9] and [10].

- The results presented herein are from actual experiment carried out in real time environment while the results of the related works mentioned in [9] and [10] are obtained from simulation environment.

International Electronic Conference on Sensors and Applications
1-16 June 2014

Conclusion

- Node localization 2D → 4 anchor nodes → MLP (feed-forward) Neural Network → 12-12-2

- An average error of 0.30 m has been achieved using 4 anchor nodes only.

- BR training method gives the best result but requires lot of time to train the network (suitable for offline methods).

- LM training method gives comparable results and suitable for online methods → efficient and requires less training time.

- Increasing the number of anchor nodes increases the localization accuracy → at the expense of higher cost

Thank you for your attention