LOCALIZATION SYSTEM FOR UAV/UGV IN URBAN ENVIRONMENTS

by

VISHAL SAVIO COELHO

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2009

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Lewis for his continual support and for giving me the opportunity to work with him and the ACS group. I am also grateful to Dr. Losh, whose courses on embedded systems provided me with a foundation in the practical aspects of system design.

I would especially like to thank Emanuel Stingu for his guidance and help without which this work would be incomplete.

The support of the Army Research Office and Army National Automotive Center in grant W91NF-05-1-0314 is gratefully acknowledged.

November 16, 2009

ABSTRACT

LOCALIZATION SYSTEM FOR UAV/UGV IN URBAN ENVIRONMENTS

Vishal Savio Coelho, M.S.

The University of Texas at Arlington, 2009

Supervising Professor: Frank Lewis

An implementation of relative localization of wireless sensor nodes using a potential field method is presented in this work. The system is designed to assist in UAV/UGV navigation in urban environments like rooms and passageways of buildings. An unmanned vehicle will need to rely on ground-based sensors to help it navigate indoors, as GPS signals are severely attenuated and reception is intermittent. A sufficient number of sensors are first deployed throughout the terrain in a random fashion. They must then localize themselves to the environment, i.e. they must develop an internal frame of reference or co-ordinate system. A potential field method is used to achieve localization, where the potential (cost) is a function of the internode distances. Each node within the network is equipped with a *Radio* and *Ultrasonic* module and the distance between nodes is calculated by measuring the difference in *time of flight* of the RF and ultrasonic signals. The radio modules, based on the IEEE 802.15.4a standard, facilitate both internode communication and two-way ranging, making them ideal for use in an ad hoc network. A PC capable base station runs the localization algorithm in MATLAB thereby reducing the computational load on the nodes. Once localized, the nodes begin tracking the vehicle

using a *Kalman* filter to estimate its trajectory. The data from the network has a limited update rate that is insufficient to track fast moving vehicles. The Kalman filter predicts the motion of the vehicle using its dynamic model and then corrects its trajectory when data becomes available. The hardware used in the sensor design was developed by the author, including electronic schematics, PCB design, component soldering and part of the supporting software.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The use of Unmanned Aerial Vehicles(UAV) and Unmanned Ground Vehicles(UGV) in military and commercial applications has risen over the past years with applications ranging from target tracking, communications relay, electronic intelligence and search and rescue to commercial use in agriculture, pollution monitoring and maritime traffic monitoring [1].

An example of the usefulness of UAV/UGV's in urban environments is search and rescue missions [2]. In an urban disaster scenario like a collapsed building, the time critical task of locating victims can be accomplished quickly through the use of autonomous vehicles which, given their size and capabilities, can maneuver through tight spaces and hazardous environments without endangering the lives of the rescue team.

A UAV/UGV requires tracking infrastructure to provide it with continual position updates, enabling it to navigate over any terrain. This is usually accomplished using Global Positioning System (GPS) signals. GPS signals, however, are unreliable in rooms and passageways of buildings as the signal is highly attenuated or reflected. A ground based sensor network that can be deployed at random prior to sending in the UAV/UGV provides a more viable approach to the tracking problem. Once a sensor network is deployed and localized it can provide accurate location and orientation information to the UAV/UGV.

A sensor network is comprised of a number of nodes, each capable of performing basic communication and ranging functions. These nodes may be randomly deployed

throughout the navigable terrain. The nodes, upon initial deployment, are unaware of the environment and their position within it. The first step in making them location aware is to create a co-ordinate system or an internal frame of reference. This process is referred to as *Localization*.

There are two categories: relative and absolute localization. In relative localization, the distance measured between nodes is used to localize them with respect to some arbitrary internal coordinate system. Absolute localization means that the nodes are aligned with an external meaningful system like GPS [3].

1.1 Localization Hardware and Ranging Methods

A prerequisite to localizing a network is the presence of *Beacon* nodes, nodes that know their location *a priori*. This knowledge may be hard coded, or acquired through the use of GPS hardware. The use of beacon nodes simplifies the task of assigning real world coordinates to ordinary nodes.

One of the more technically challenging hardware problems in the localization process is calculating the distance between sensors. Various approaches involving RSSI, TOA, AOA and signal pattern matching can be used to tackle this problem [4]. In many wireless sensor networks, nodes are equipped with short range low power radios. They can serve a dual purpose, allow nodes to communicate with each other as well as determine the distances of separation between them. There are two methods of ranging using radios, one using *Received Signal Strength Indication (RSSI)*, the other hop count.

The first method (RSSI) works on the basis that radio signal strength diminishes with the square of the distance from the signal source. If the relationship between signal attenuation and distance in a given environment is accurately known, a receiving node can determine its distance from the transmitting node using the strength of

the received signal. The signal strength does not follow a strict mathematical relationship with distance in all environments. Humidity, temperature, presence of water bodies and reflective objects affect the signal strength and so *a priori* knowledge of the environment is required to effectively employ such a solution. One such implementation using RSSI is described in [5] .The second method (hop count) is used to calculate inter-node distances in large sensor networks [6, 7]. The key idea is that if two nodes are able to communicate they must be at a distance less than R, R being the maximum communication range, with high probability. The connectivity matrix (all nodes) can then be used to localize the network. Since distance measurements are always integral multiples of a single hop distance, the accuracy degrades with each additional node with a total error of about 0.5R per measurement.

Infrared systems have been developed for indoor UAV navigation [8]. The tags use line of sight detection of infrared signals transmitted from multiple beacons to determine their position and orientation; however, they are equipped with IR cameras and their associated motion systems which make them bulky and power hungry. They are not portable and impractical in scenarios requiring quick and easy setup. Another approach similar to the IR system would be to use IR LEDs or microphone arrays to determine the direction of a transmitting node and thus calculate inter-node distances using *Angle of Arrival(AoA)* data as described in [9], however such an implementation would make the sensors bulky and expensive.

A commonly used hardware scheme for ranging is *Time Difference of Arrival (TDOA)*. In TDoA systems, each node is equipped with a radio and an ultrasonic sensor. The transmitting node will first send an RF chirp signal followed by an ultrasonic pulse train. When the receiving or listening node hears the radio signal, it notes the current time and immediately turns on its own ultrasonic sensor. When the sensor picks up the ultrasonic pulse train, it notes the time. The difference in

time between the reception of the radio signal and the ultrasonic signal can be used to calculate the distance between nodes, given the exact speed of RF and Ultrasonic transmissions in air. Performance analysis and techniques to improve the accuracy of TDOA measurements in wireless sensor networks is described in [10] . The Cricket System developed at MIT uses this particular approach to ranging [11]. The network uses a set of fixed nodes placed at regular intervals to track a mobile node. Each node possesses an array of ultrasonic receivers to determine a tag's (target) orientation in addition to its location in the environment. The system is being developed to promote pervasive computing in urban environments, where a user is able to seek out resources and interact with them wirelessly [12].

1.2 Problem Description

Most implementations of Wireless Sensor Networks (WSN) in an urban environment are geared towards developing smart environments [13] that respond to sensory data from the real world. They require a fixed infrastructure i.e. the nodes must be placed strategically to maximize coverage and minimize the total number of nodes. An example of such an implementation used to monitor a heating and air conditioning plant is given in [14]. This approach will not work in a surveillance or search and rescue operation which requires the network to be deployed on demand. The objective of this work is to implement such a system. The nodes that constitute the network can be deployed at random to form an ad hoc network. They will then localize themselves to the environment and provide data which can be used by a UAV or UGV to carry out its mission objectives.

The sensors were custom designed and built by the author for this specific implementation. Each node in the network is equipped with an IEEE 802.15.4a capable radio module and a MaxBotix ultrasonic sensor shown in Fig 1.1. They

utilize the TDoA approach to ranging as it provided the most cost effective and power efficient solution to develop and deploy.

Figure 1.1. Radio and Ultrasonic Modules.

The Radio modules themselves have an in-built ranging functionality employing a technique called *Symmetric Double Sided Two Way Ranging (SDS-TWR)* [15]. It is a Time of Flight method that works on a two-way handshake protocol, wherein the reader (node that initiates a ranging operation) will send a signal and wait for an acknowledgement from the tag (target node). Once complete, the roles of reader and tag are switched and the same operation is repeated. Each signal has a time stamp embedded in it which allows both the tag and reader to calculate the distance to one another, given the speed of RF signals. The radios provide reasonably accurate measurements of within a meter. The use of ultrasonic sensors in tandem with the radios allows for greater resolution of up to about a few centimeters, which is desired when navigating through close quarters.

The relatively slower speed of sound means that measurement updates from the network are limited by the time it takes for an ultrasonic signal to travel between nodes (a few milliseconds). The slow update rate makes it difficult to actively track

a fast moving object like a UAV. The problem can be overcome by using a *Kalman filter* [16, 17] to predict the motion of the target vehicle using its dynamics and then correcting those estimates when new data becomes available.

The following chapter details the system description and mathematical framework. Chapter 3 discusses the hardware organization and implementation. Chapter 4 evaluates the sensor network in different environments and finally chapter 5 summarizes the results and conclusions arrived at and also talks about further improvements that can be made to the system.

CHAPTER 2

ORGANIZATION OF THE SENSOR NETWORK

The sensor network is comprised of n wireless sensor nodes, of which n_1 nodes make up the ground based network (shown in Fig 2.1)and the remaining n_2 nodes are placed on the target vehicles to be tracked. The nodes can be placed at random within a room with the single constraint that each node is in communication range with at least two other nodes. Ranging is accomplished through the combined use of the radio transceivers and ultrasonic sensors. The time of travel between the nodes of both the RF and the ultrasonic signals is measured. The ultrasound data provides a more precise measurement, but it only works for small distances (6 meters).The radio data has a lower resolution, but it is available for larger distances (60 meters). The mobile robots that are equipped with the localization nodes can use lower-resolution position data when they are far away from each other and start using both radio and ultrasound when they are in close proximity for finer maneuvers. All range information is communicated to a PC running the localization algorithm. A GUI, created in MATLAB, displays the results of localization and the tracking operation in real time. A depiction of the setup is shown in Fig 2.2.

The Printed Circuit Board (PCB) of the sensor node is shown in Fig 2.3. The ultrasonic sensor and associated analog processing circuitry lie on the front side of the node. A dsPIC33FJ128GP802 microcontroller from Microchip, placed beneath the ultrasonic sensor, controls the ranging and communication operations of the node. The radio transceiver, serial communication, power distribution block and program-

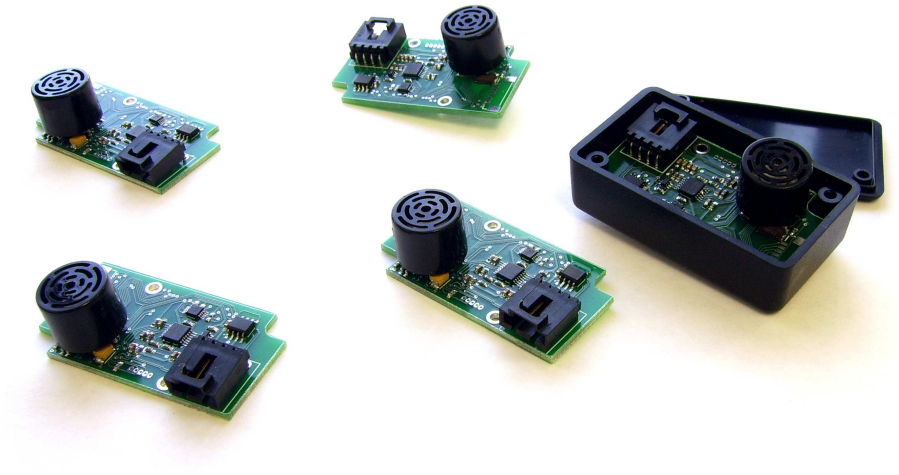

Figure 2.1. Ground Based Sensor Nodes.

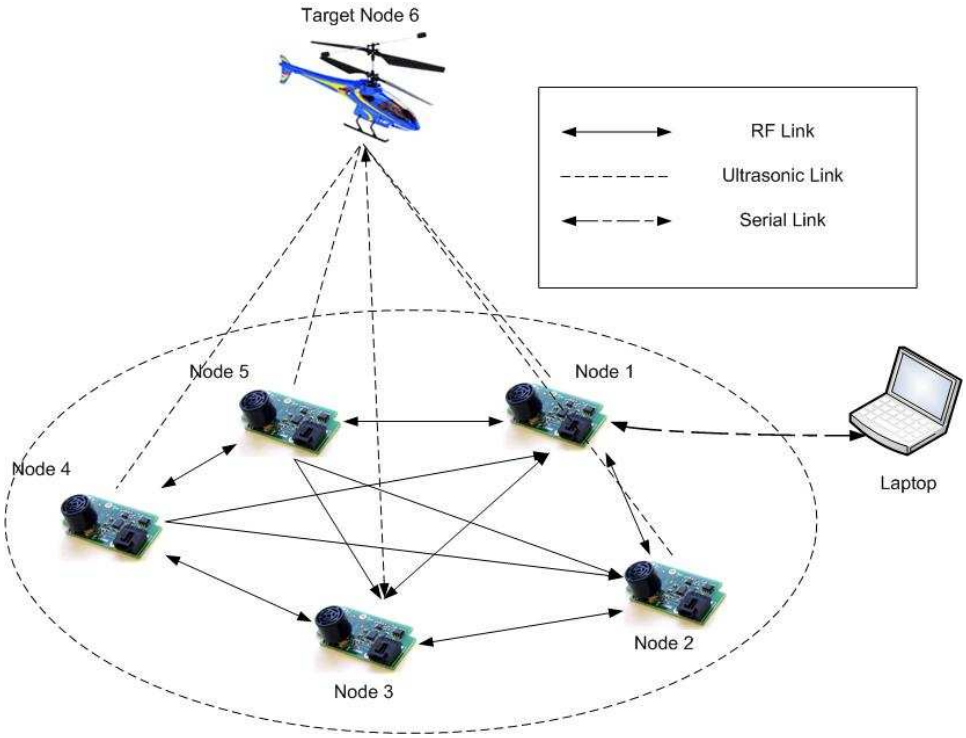

Figure 2.2. Organization of the Sensor Network.

ming ports lie on the reverse side of the PCB. The form factor of the board is designed to fit in a Hammond 1551H box.

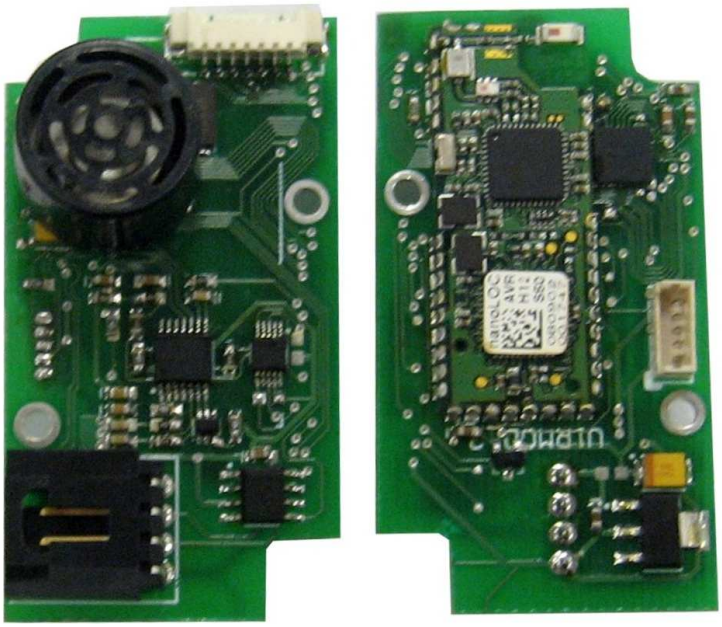

Figure 2.3. Front and reverse sides of the sensor node.

2.1 Radio Transceiver

The radio transceivers are commercially available nanoLOC AVR modules developed by NanoTron Technologies Inc. They are compliant with the IEEE 802.15.4a communications standard and have an additional ranging functionality built in. The transceivers operate in the 2.450 GHz ISM (Industrial, Scientific and Medical) band. Each module has an NA5TR1 transceiver IC controlled by an ATMega644V(Atmel) microcontroller. Of specific interest is the ranging capability of these transceivers. They use a ranging methodology developed by Nanotron called *Symmetrical Double-Sided Two-Way Ranging (SD-TWR)*. It is *symmetrical* in that the measurement from

a local nanoLOC station to a remote nanoLOC station is mirrored by a measurement from the remote station to the local station (ABA BAB). It is *Double-Sided* in that only two stations are used to get range measurements and *Two-way* because a data packet sent from one station is acknowledged by the other. This concept is illustrated in Fig 2.4.

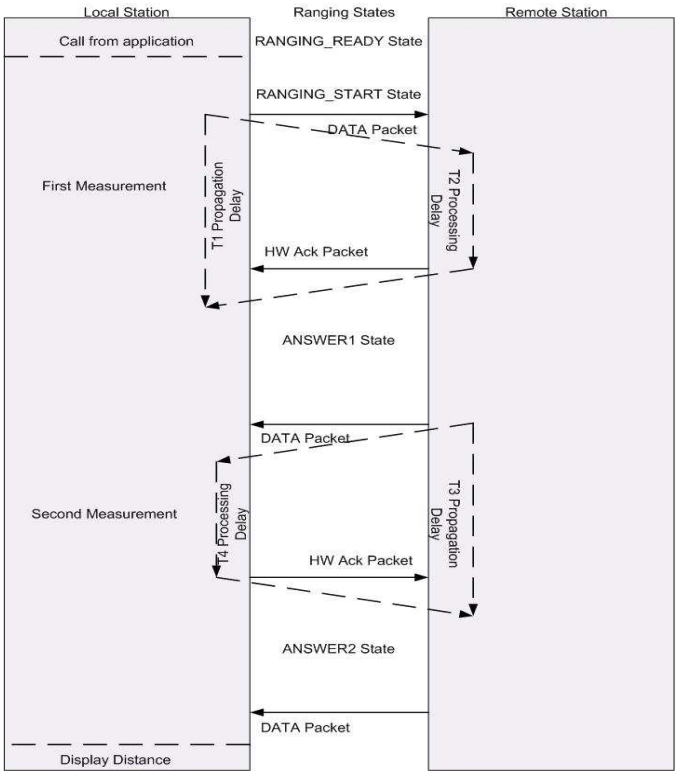

Figure 2.4. Front and reverse sides of the sensor node.

A call from the user application initiates the ranging operation. Time stamps are embedded in each data packet sent across stations, which are used to record the processing delays between responses. Once complete, two sets of time values

(ANSWER1 and ANSWER2) are used to calculate the range using the following formula:

$$Distance = \frac{(T_1 - T_2) + (T_3 - T_4)}{4} \times c \tag{2.1}$$

where,

$T_1 - T_2 = ANSWER1$

T_1 = propagation delay time of a round trip between a local and remote station

T_2 = processing delay in the remote station

$T_3 - T_4 = ANSWER2$

T_3 = propagation delay time of a round trip between a remote and local station

T_4 = processing delay in the local station

$c = 3 \times 10^8$m/s, propagation speed of EM waves

2.2 Ultrasonic Sensor

The ultrasonic sensors mounted on the front side of the PCB have a transmission cone angle of 40 degrees. In a ranging operation 13 pulses are transmitted at a frequency between 38 - 42 KHz by a transmitting node. The target node has its ultrasonic sensor keyed to pick up the first set of pulses it finds. Its associated analog circuitry is designed to filter, amplify and integrate the received pulse train and provide a voltage pulse at its output which can be used as a crude logic signal by a microcontroller. The microcontroller can decide if a pulse was detected if the voltage level at the output of the analog circuitry exceeds a predetermined threshold voltage level (Fig 2.5).

The first opamp acts as a band pass filter picking up signals in the range of 38 - 42 KHz, filtering out much of the ambient noise thereby reducing the noise floor considerably. The second forms an inverted signal amplifier with variable gain, whose

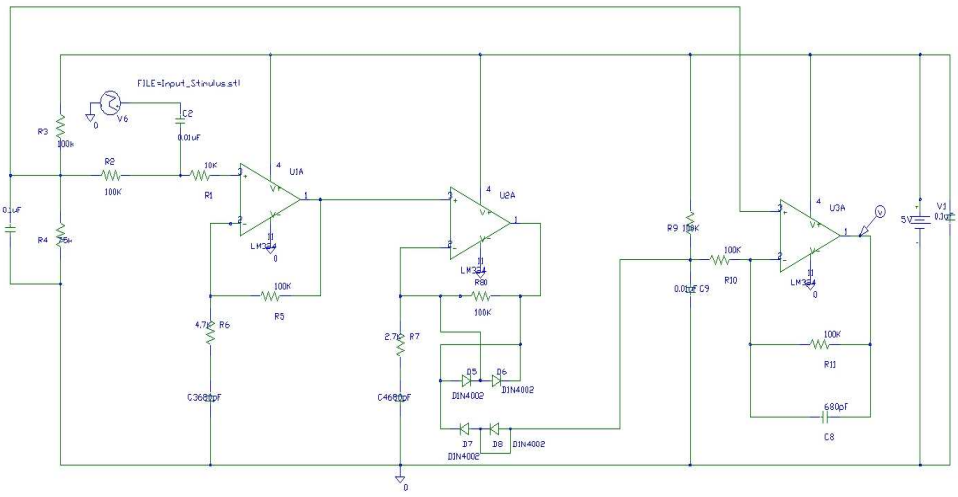

Figure 2.5. Analog circuit to process received ultrasonic pulse train.

function is to amplify the filtered signal and pass it on to the final opamp which will invert and integrate it to yield a sharp pulse at its output whenever a pulse train is detected. The signals at the input and output of the final stage integrator are shown in Fig 2.6 and Fig 2.7 respectively. The data was captured using a digital oscilloscope during an ultrasonic module test using two sensor nodes placed a few inches apart.

A problem with this basic approach to sensing is that as the distance travelled by the ultrasonic signal increases, its strength decreases. In order to tackle the problem, an SPI programmable potentiometer is placed in the feedback loop of the second stage log amplifier. The microcontroller's logic circuit will determine the feedback resistance, based on past values, that will maintain a constant amplitude at the output of the opamp. The use of variable gain makes the system robust, as it ensures that weak signals transmitted over long distances are amplified sufficiently to be detected. The microcontroller constantly samples the signal, calculates its mean and standard deviation and sets the threshold to the mean plus five times the standard deviation. The statistical approach ensures that noise spikes will not be confused for the real signal itself as the threshold is always well above the noise floor.

Figure 2.6. Filtered, amplified inverted output of the second stage amplifier.

Figure 2.7. Final output provided to the microcontroller.

2.3 Real Time Control

The dsPIC33 microcontroller running FreeRTOS v5.0.2 controls the majority of the node operations (Fig 2.8) like calibration of the ultrasonic sensors and the ranging protocol. Communication between the radio, control board and the PC is serial (UART).

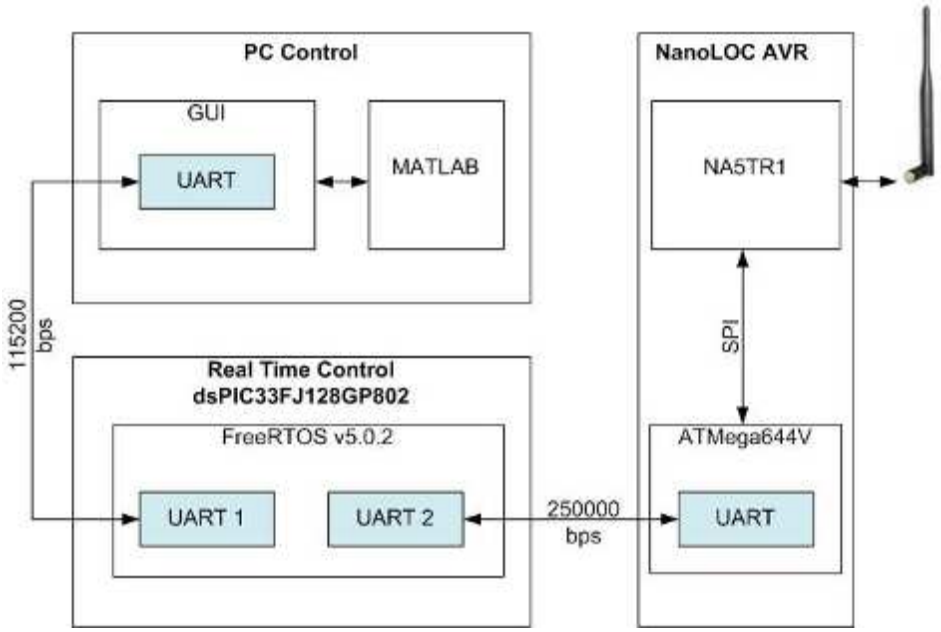

Figure 2.8. Organization of the node control system.

The control flow of the ranging cycle is depicted in Fig 2.9. There are twelve states in total and the transition between them is periodic and not dependent on any external condition.

This periodic transition or sample time is maintained using hardware timers. The system is currently setup to accommodate eight nodes only with each node given a particular time slot for transmission. A *Master* node will synchronize the other nodes by broadcasting a *SYNC* packet which allows the rest of the nodes to start

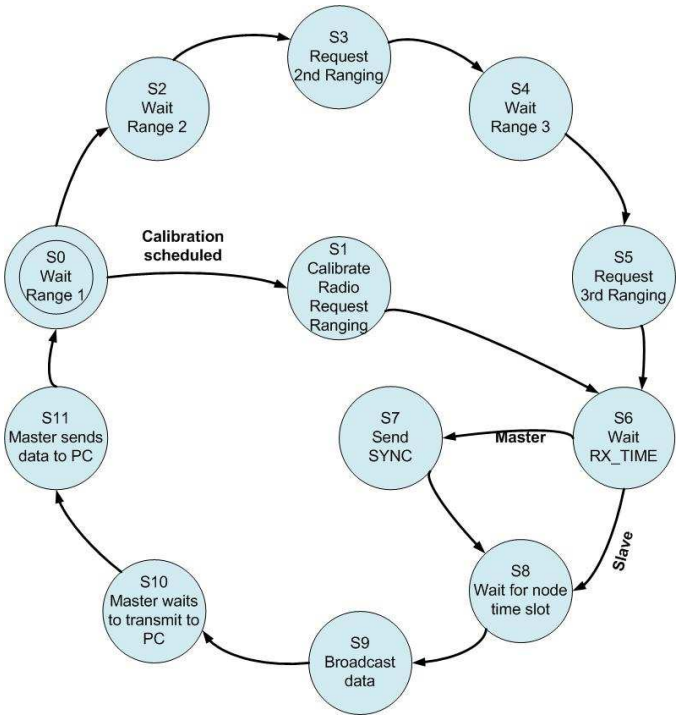

Figure 2.9. Organization of the node control system.

their sampling timers at relatively the same time. Based on the SYNC packets, the nodes estimate the timing of the Master node and can function even if they miss SYNC packets for 1 - 2 seconds. Once synchronization is achieved, the ranging operation can begin. Nodes are assigned a number or ID between 0 to 7, 0 being the *Master* node. Radio communication and localization is done in specific time slots during a 50 ms frame for each node. During each 50 ms frame an ultrasonic localization and three radio localizations are done. The distance between a node and every other node in the network needs to be determined for the algorithm to produce the most accurate results. In the first ranging cycle each node will find the distance to its immediate neighbor (ID). In the second it will find the distance to the second closest ID and in the third, to the third closest ID and so on with each ranging cycle, as shown in Fig 2.10. This ensures equitable measurement updates for all nodes. In state 0, the node

will wait for a radio range measurement that was started during the previous ranging cycle. It will then calibrate its radio and request the first ranging measurement during state 1.The next five states correspond to a sequence of computed delays and range measurement request. The node will initiate a delay during state 8 until it has reached its particular time slot and then broadcast its sensor data to all the other nodes (state 9). States 10 and 11 are relevant to the master node which will transmit data it received to the PC via the serial port.

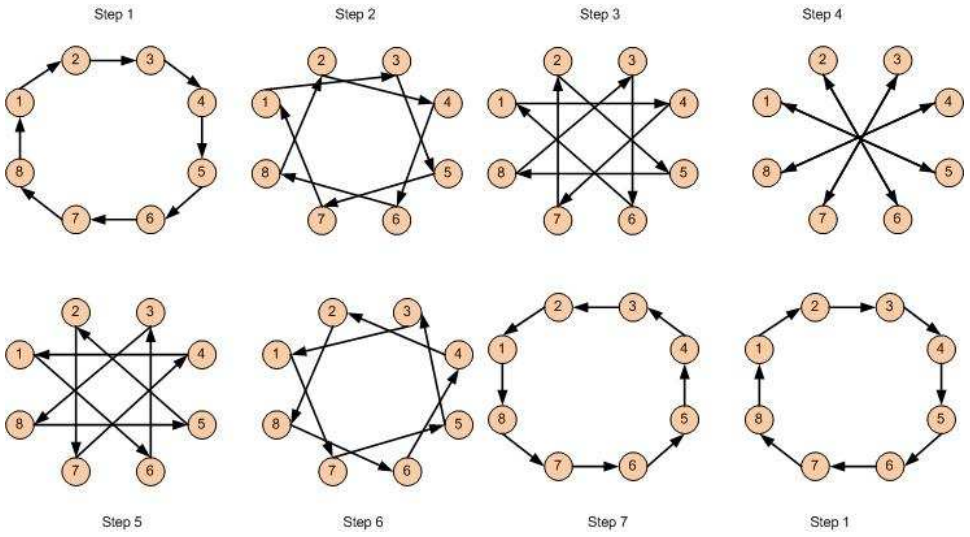

Figure 2.10. Ranging operation in steps.

Each of the nodes is equipped with a single ultrasonic sensor with a cone angle of approximately 40 degrees limiting the area of coverage. Since the nodes are placed on the ground facing up, the ultrasonic signals are bounced off the ceiling to other nodes while ranging. The height of the room at a particular node is determined by performing a self ranging operation, where a node picks up its own ultrasonic reflection from the ceiling. The distance between nodes forms the base of an isosceles triangle (Fig 2.11) and is determined using basic geometry.

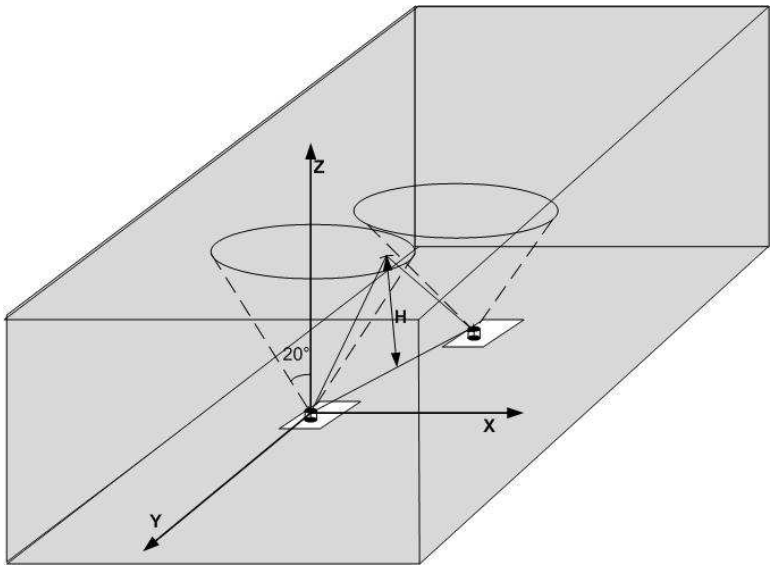

Figure 2.11. Calculation of inter-node distance.

In the Ranging state, each node takes its turn to transmit while the other nodes listen. At the beginning of the 50 ms frame, the active node transmits 13 ultrasonic pulses at a frequency of 41.666 KHz. Each node starts a timer at the start of the frame. Once the pulse train is detected at each receiving node, the receiving nodes stop their timers and begin calculating their distance from the transmitting node using the recorded time of flight data. In a similar fashion the next node begins transmitting while the other nodes listen. This process continues till all the nodes in the ground based network have finished transmitting in sequence and inter-node distances have been calculated and transmitted to the other nodes, of which some can relay the data to the PC. The round robin approach to ranging obviates the need to have complicated collision avoidance algorithms.

The localization algorithm (explained in chapter 3) will develop an internal co-ordinate based on the inter-node distances obtained from the sensor network making it possible to track a target UAV/UGV. In the tracking phase the mobile nodes need to be in range of at least three stationary nodes to get a fix on position. In the event

that the data is unavailable the *Kalman* filter predicts the position of the target node based on the last good measurement and the target's dynamic model. When new data is available the filter corrects for errors in its estimate and refines its parameters for the next set of estimates. This is done in real time with node 0 acting as the point of contact between the real time algorithm on the PC and the sensor network.

2.4 PC Software

The localization algorithm is computationally intensive and therefore infeasible to be run on the sensor nodes. For this reason, both the localization and tracking algorithms are run on a PC using MATLAB. A GUI receives telemetry data from node 0 of the ground network through the serial port and also continuously monitors and displays the status of the network. It allows the user to re-localize the modules if they are deployed at a different location. The GUI displays the data in real time (shown in Fig 2.12) with updates from the sensor network available 20 times a second.

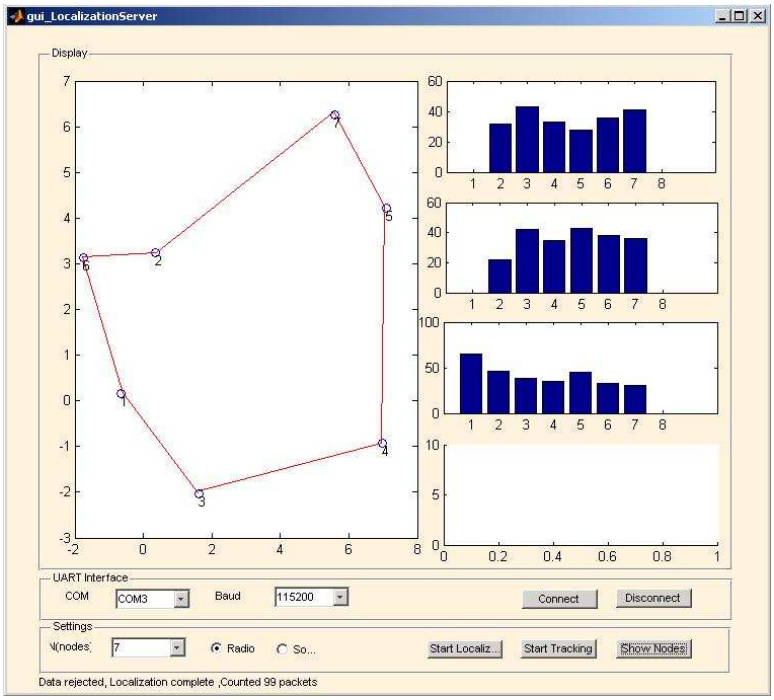

Figure 2.12. Graphical User Interface.

CHAPTER 3

SYSTEM DESCRIPTION

The method of localization uses a dynamical model to determine node position estimates by driving each node using a fictitious virtual force based on the range error [18]. In the state space description of the dynamics, a few of the states are the position estimates of the nodes. The system attempts to reach a steady state value that is optimal in a least- squares sense. It should be noted that the nodes do not move physically, but that the frame of reference is recalculated each time a new node is added to the network giving the impression that the nodes are shifting.

A description of the system of equations used to generate position estimates based on inter-node distance information available from the sensor network is described here. Let initial position estimates of a node i, based on the available range information be

$$X_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T \tag{3.1}$$

with position estimate dynamics

$$\ddot{X}_i = f_i \tag{3.2}$$

where

$$f_i = \left[\vec{f_i^x}\, \vec{f_i^y}\right]^T$$

are the virtual forces along the x and y directions respectively. To obtain a linear representation of the second order system we assume two *states* of the system, namely x_1 and x_2, to be as follows,

$$x_1 = X_i \text{ and } x_2 = \dot{X}_i$$

or

$$x_1 = \begin{bmatrix} x_i & y_i \end{bmatrix}^T$$

and

$$x_2 = \begin{bmatrix} \dot{x}_i & \dot{y}_i \end{bmatrix}^T$$

Therefore,

$$\dot{x_1} = x_2 \text{ and } \dot{x_2} = \vec{f_i}$$

The state space model is therefore,

$$\begin{bmatrix} \dot{x_i} \\ \dot{x_i} \end{bmatrix} = \begin{bmatrix} \dot{x_i} \\ \dot{y_i} \\ \ddot{x_i} \\ \ddot{y_i} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ \dot{x_i} \\ \dot{y_i} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} f_i^x \\ f_i^y \end{bmatrix} \tag{3.3}$$

3.1 Potential Field for Optimal Position Estimate

We introduce a potential field to determine the virtual forces along both the x and y directions (Eq 3.2) so that the position estimates reach a steady state value that is optimal in the least square sense. The field is defined as

$$V_{ugs} = \sum_{i=1}^{N} \sum_{j=1, i \neq j}^{N} \frac{1}{2} K_{ij} \left(r_{ij} - \bar{r}_{ij} \right)^2 \tag{3.4}$$

where $i \neq j$,$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ is the estimated distance between nodes i and j and $\bar{r_{ij}}$ is the measured distance between nodes i and j. The potential function for a single node i is given by

$$V_{i_{ugs}} = \sum_{j=1, i \neq j}^{N} \frac{1}{2} K_{ij} \left(r_{ij} - \bar{r}_{ij} \right)^2 \tag{3.5}$$

The gradient, i.e., the change in potential of a single node with respect to its states is given as

$$\frac{\partial V_{i_{ugs}}}{\partial X_i} = \vec{\nabla} V_{i_{ugs}} = \vec{\nabla} \sum_{j=1, i\neq j}^{N} \frac{1}{2} K_{ij} \left(r_{ij} - \bar{r}_{ij}\right)^2 = \sum_{j=1, i\neq j}^{N} \frac{1}{2} K_{ij} 2 \left(r_{ij} - \bar{r}_{ij}\right) \frac{\partial \left(r_{ij} - \bar{r}_{ij}\right)}{\partial X_i} \tag{3.6}$$

where,

$$\frac{\partial \left(r_{ij} - \bar{r}_{ij}\right)}{\partial X_i} = \frac{x_i - x_j}{\parallel x_i - x_j \parallel} \hat{x} + \frac{y_i - y_j}{\parallel y_i - y_j \parallel} \hat{y} \tag{3.7}$$

Therefore, from Eq 3.6 and Eq 3.7 we get

$$\frac{\partial V_{i_{ugs}}}{\partial X_i} = \sum_{j=1, i\neq j}^{N} K_{ij} \left(r_{ij} - \bar{r}_{ij}\right) \left[\frac{x_i - x_j}{\parallel x_i - x_j \parallel} \hat{x} + \frac{y_i - y_j}{\parallel y_i - y_j \parallel} \hat{y} \right] \tag{3.8}$$

Let the force for the i^{th} node be

$$\vec{f_i} = - \sum_{j=1}^{N} K_{ij} \left(r_{ij} - \bar{r}_{ij}\right) \frac{X_i - X_j}{\parallel X_i - X_j \parallel} - K_v \dot{X}_i \tag{3.9}$$

Double integration of the state estimator given in 3.2 yields a steady state value for the estimated position of the node which is the best fit given the data available at the time of integration. The estimates is considered optimal in the sense that the cost function V_{ugs} is minimized. The proof is beyond the scope of this work and a more detailed analysis is provided in [18].

3.2 Relative Localization

A minimum of three nodes are needed to calculate a unique starting point for the coordinate system. Node 1 is assumed to be the origin with co-ordinates ($x_1 = 0$ $y_1 = 0$). Node 2 lies along the x-axis at a distance r_{12} from node 1 ($x_2 = r_{12}$ $y_2 = 0$). The addition of a third node completes the triangle (shown in Fig 3.1) and its co-ordinates (x_3,y_3) can be calculated using simple geometric principles;

$$x_3 = r_{13} cos(\theta); y_3 = r_{13} sin(\theta)$$

where θ can be determined using the law of cosines.

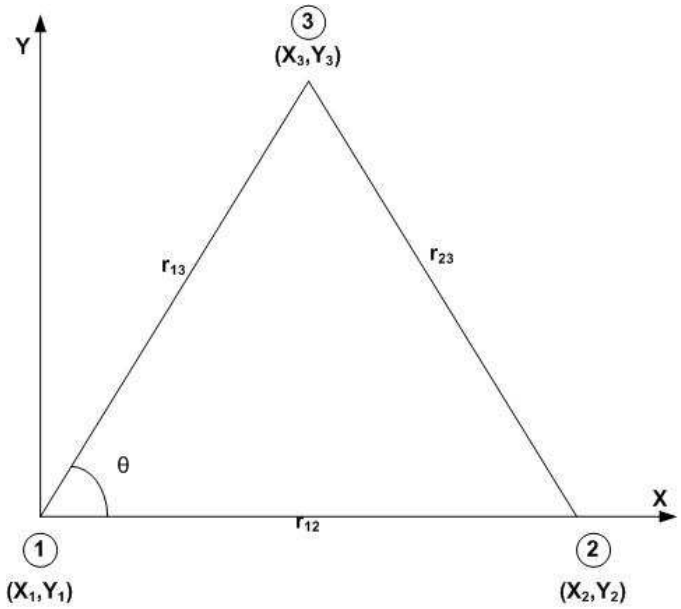

Figure 3.1. Initial setup with three nodes.

For every subsequent node j, we can determine an initial starting position using the method of trilateration [19]. Consider Fig 3.2 which shows the process of trilateration being used to determine the starting position for node 4. Three circles are constructed centered about nodes 1, 2 and 3 with radii equal to their linear distance from node 4. The starting position of node 4 is determined using the following equation

$$\begin{bmatrix} x_4 \\ y_4 \end{bmatrix} = \begin{bmatrix} 2(x_1 - x_3) & 2(y_1 - y_3) \\ 2(x_2 - x_3) & 2(y_1 - y_2) \end{bmatrix}^{-1} \begin{bmatrix} \bar{r}_{34}^2 - \bar{r}_{14}^2 + x_1^2 - x_3^2 + y_1^2 - y_3^2 \\ \bar{r}_{34}^2 - \bar{r}_{24}^2 + x_2^2 - x_3^2 + y_2^2 - y_3^2 \end{bmatrix}$$

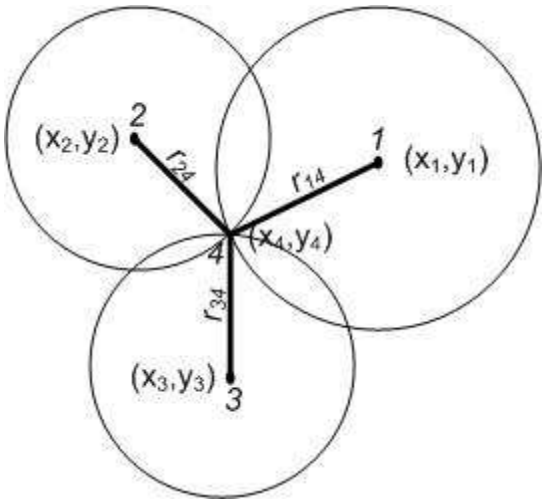

Figure 3.2. Trilateration.

A generalization of the above equation to determine the starting position of a new node j is given as

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = \begin{bmatrix} 2\left(x_{j1} - x_{j3}\right) & 2\left(y_{j1} - y_{j3}\right) \\ 2\left(x_{j2} - x_{j3}\right) & 2\left(y_{j1} - y_{j2}\right) \end{bmatrix}^{-1} \begin{bmatrix} \bar{r}^2_{j3j4} - \bar{r}^2_{j1j4} + x^2_{j1} - x^2_{j3} + y^2_{j1} - y^2_{j3} \\ \bar{r}^2_{j3j4} - \bar{r}^2_{j2j4} + x^2_{j2} - x^2_{j3} + y^2_{j2} - y^2_{j3} \end{bmatrix} \tag{3.10}$$

where, (x_j,y_j) are the position estimates for the j^{th} node (x_{j1}, y_{j1}),(x_{j2}, y_{j2}) and (x_{j3}, y_{j3}) are the position estimates of the already localized nodes j_1,j_2 and j_3 respectively $\bar{r}_{j_i j_k}$ is the measured distance between nodes j_i and j_k

Once the estimate of the position for the newly introduced node is available, the network is localized by applying the virtual forces given in Eq 3.9 as a control input for each node i belonging to the network. This allows the relative position estimates of all the nodes in the network to be adjusted each time a node is added.

3.3 Ranging

The ultrasonic signal is the strongest along the axis normal to its own surface and starts to decrease exponentially as it moves away from the normal with virtually

no energy being channeled along its surface. Assuming the nodes are placed flat on the ground with their ultrasonic sensors facing upward, the distances measured between nodes will be that of path traced by the signal reflected off the ceiling. The sensors would have to be extremely close for a direct signal to be detected and such a placement would not be of much use in a tracking scenario. The distance between nodes forms the base of an isosceles triangle with the equal sides forming the path that the ultrasonic signal takes from the source node to the ceiling and then finally to the destination node. The height of the room at a node's position is given by

$$H_i = T_r \times V_s \tag{3.11}$$

where H_i = Height of the room at node i's position

T_r =Time it takes for 13 ultrasonic pulses to travel from the node to the ceiling and back

V_s =Speed of sound in air at room temperature = 346.65 m/s

Consider two nodes, 1 and 2 at a distance d_{12} from each other as shown in Fig 3.3. Node 1 will transmit a chirp signal followed by an ultrasonic pulse train while node 2 listens for both signals marking their arrival with the use of in-built timers. If T_{12} is the time it takes for the ultrasonic signal to travel from 1 to 2, then the distance between the two nodes is calculated as follows

$$d_u = T_{12} \times V_u \tag{3.12}$$

where d_u = Distance covered by the ultrasonic signal when travelling from node 1 to 2

Using Pythagoras' theorem we can determine the distance between nodes

$$d_{12} = 2 \times \sqrt{\frac{{d_u}^2}{2} - H^2} \tag{3.13}$$

where H = Height of the room determined at the receiving node 2

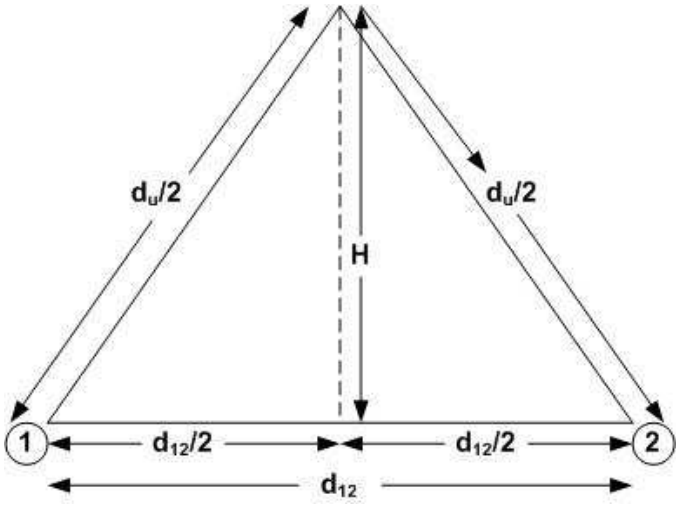

Figure 3.3. Determining the distance between two nodes.

3.4 Kalman Filter

The measurement updates in the system are available every 50 milliseconds or 20 times a second. Assuming the heading or angular velocity of the vehicle is not available to us, it becomes difficult to track the position of a fast moving target very accurately. In such cases it becomes necessary to know the model or dynamics of the vehicle to be able to predict its movement over time. The Kalman filter has two phases of operation, a temporal update phase and a measurement update phase. In the temporal update phase the filter predicts the states (position and heading) of the target vehicle using its dynamic model and then corrects errors in its prediction during the measurement update phase using the data it acquired from the sensor network.

Consider the model of a simple ground vehicle [20]. It has 3 states: the position along the x and y axes and the angle it makes with the positive x axis (heading).The inputs to the system are the measured velocity (V) and angular velocity (ω). The dynamic model of the system is non linear and can be represented by the following three equations

$$\dot{x} = V.Cos\theta \tag{3.14}$$

$$\dot{y} = V.Sin\theta \tag{3.15}$$

$$\dot{\theta} = \omega \tag{3.16}$$

Since the system is non linear, an Extended Kalman Filter (EKF) [21] is used. The system of equations can be linearized by taking the Jacobian or partial derivative with respect to the states and evaluating them at points along a predicted trajectory. The equations necessary to predict the states of the vehicle are given below.

$$\hat{x}_k(-) = \hat{x}_{k-1}(+) + \int_{t_k}^{t_{k-1}} f(\hat{x}, t)dt \tag{3.17}$$

$\hat{x}_k(-)$ is the predicted or apriori state of the system while $\hat{x}_{k-1}(+)$ is the aposteriori or corrected prediction of the states. $f(\hat{x}, t)$ represents the nonlinear dynamics of the system.

The matrix of the covariances of the predicted states provides a measure of uncertainty in the estimation process. The uncertainty could be due to disturbances in the environment. The covariance matrix varies over time and is expressed by the following equation,

$$\dot{P} = FP + PF^T + Q(t) \tag{3.18}$$

where P is the covariance matrix of the prediction estimates, F is the Jacobian of the non linear function $f(\hat{x}, t)$ and Q is the covariance matrix of the dynamic disturbance in the system

The next set of equations describe the measurement update process where each new set of data points refines the estimated trajectory of the vehicle.

$$\hat{x}_k(+) = \hat{x_k}(-) + \bar{K}[z_k - h_k(\hat{x_k}(-))] \tag{3.19}$$

where $\hat{x}_k(+)$ is the corrected estimate of the states, z_k represents the measurement at time k, $h_k(\hat{x_k}(-))$ is the non linear function that describes how the states are related to the predicted measurements and $\bar{K}$ is the Kalman gain factored which is obtained by solving the Riccati equation,

$$\bar{K} = P_k(-)H_k^T[H_k P_k(-)H_k^T + R_k]^{-1} \tag{3.20}$$

where $P_k(-)$ is the covariance matrix of the prediction estimates prior to a measurement update, H_k is the Jacobian of the measurement sensitivity function $(h_k(\hat{x_k}(-)))$.

The filter provides a robust way of accurately tracking vehicles when the data in the system is unreliable.

CHAPTER 4

SYSTEM EVALUATION

An example of a test scenario is shown in Fig 4.1. There are five nodes numbered 1 through 5 that comprise the ground based network while a sixth node is placed on the target vehicle (in this case, a UGV). The ultrasonic sensors have a range equal to $4H\tan(20°)$ m, where H is the height of the room at the position of the node. In order to have finer resolution in terms of range measurements, each node must be placed within ultrasonic range of every other node. At distances exceeding the ultrasonic sensor range of 3.5 meters, assuming the standard height of a room is 2.43m, the nodes will have to rely solely on range measurements obtained from the radios which have a minimum resolution of 1m, providing coarse tracking data.

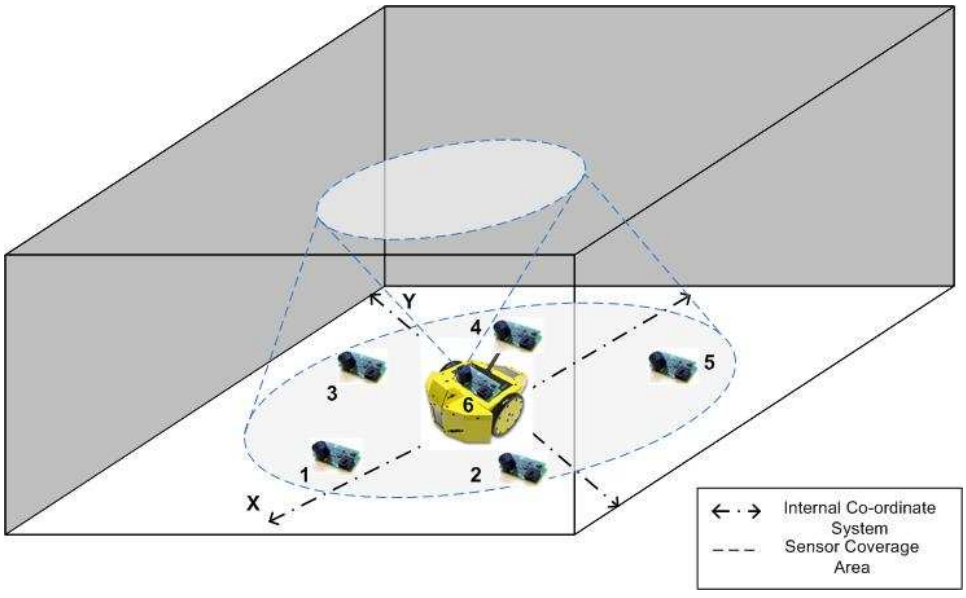

Figure 4.1. Test Setup.

After the nodes calibrate their radios, they must determine their distance from the other nodes. In a simulation run, the position of 5 nodes with randomly distributed noise added were generated as shown in Fig 4.2 and the distances between them were calculated to give the values in Table 4.1.

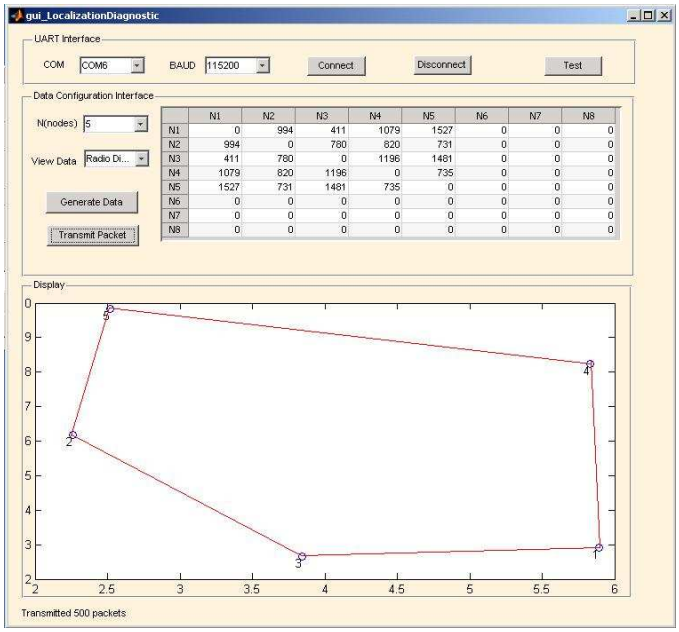

Figure 4.2. Test Setup.

Table 4.1. Generated Range Data with random noise

Node	1	2	3	4	5
1	0.00	4.88	2.06	5.34	7.70
2	4.88	0.00	3.85	4.12	3.66
3	2.06	3.85	0.00	5.93	7.29
4	5.34	4.12	5.93	0.00	3.67
5	7.70	3.66	7.29	3.67	0.00

The localization process works as follows: The initial positions of nodes 1, 2 and 3 are assumed to be random integers in the range [0,1] for both the x and y co-ordinates. The second order differential equation describing the dynamics is evaluated using the ode15s (ordinary differential equation variable order solver) function in MATLAB to give the optimal position estimates for nodes 1, 2 and 3 (shown in Fig 4.3). The GUI will track the movement of the nodes from their initial to final position estimates. The nodes do not physically move, rather the algorithm tries to find the best fit, in the least squares sense, for the position of the node using the available data. It tries to minimize the error between its prediction and the data it receives from the sensors using the potential field method described in Chapter 3. It gives the appearance of the nodes shifting position.

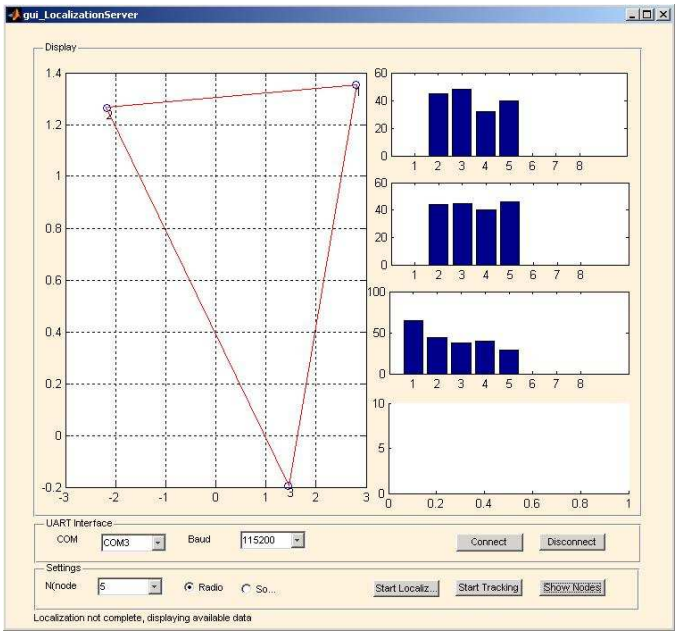

Figure 4.3. Corrected positions for nodes 1, 2 and 3.

When a new node is added to the system, the existing nodes exert a virtual force on it. The force is a function of the error in prediction of the position of the

new node. The force exerted along a particular direction is greater if the error in the calculated distances between nodes along that direction is greater. The algorithm tries to converge to the position of least error in the quickest possible time using these forces. The ode15s function is used to solve the dynamics for each new node to obtain its optimal position. Fig 4.4 and Fig 4.5 show the optimal positioning of nodes 4 and 5 while 4.6 depicts the final optimal position estimates for all 5 nodes in the network. The system exits the localization state and begins actively tracking the target vehicle until commanded otherwise.

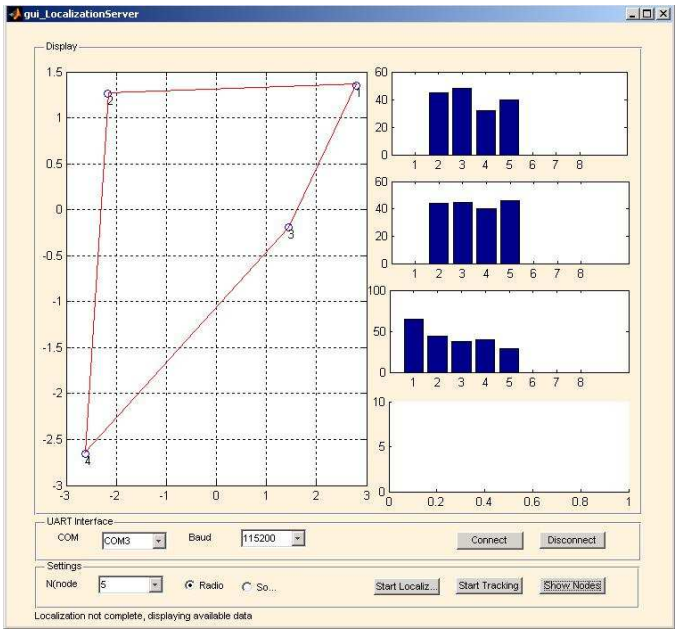

Figure 4.4. Optimal position estimates of the 4^{th} node.

After the localization of all nodes is complete, the ranges were re-calculated using the optimal position estimates of the nodes. The results are shown in Table 4.2. There were deviations in the post localization inter-node distances from the

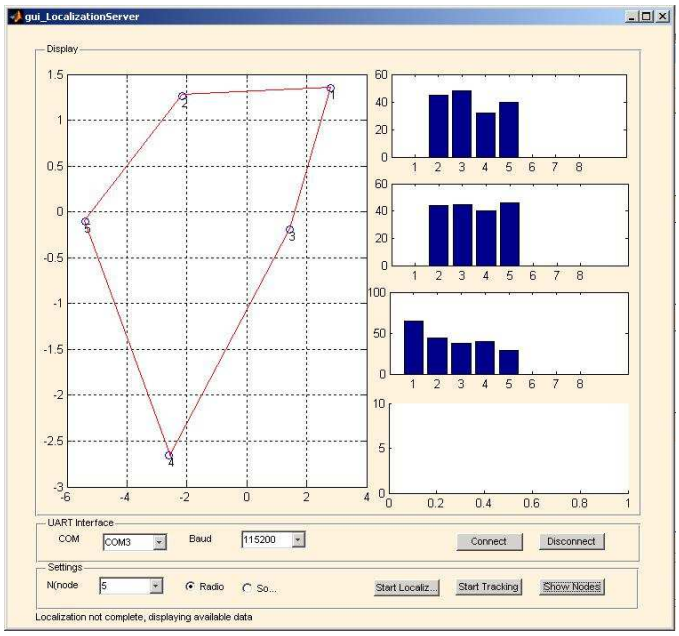

Figure 4.5. Optimal position estimates of the 5^{th} node.

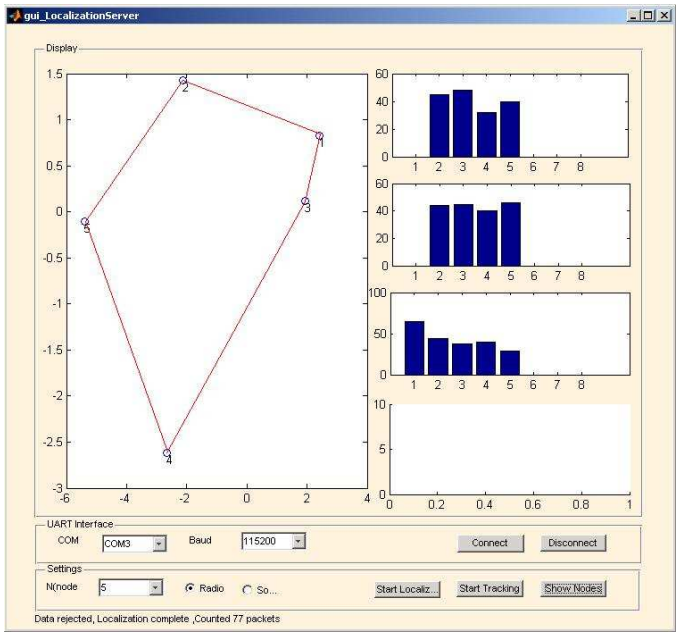

Figure 4.6. Optimal position estimates of all 5 nodes.

measured distances given in Table 4.1 .The percentage relative error in range data was calculated as follows

$$RelativeError(\%) = \frac{PostLocalizationRangeValue - MeasuredRangeValue}{MeasuredRangeValue} \times 100$$

The results are shown in Table 4.3.

Table 4.2. Post Localization Range Data

Node	1	2	3	4	5
1	0.00	4.57	0.86	6.12	7.85
2	4.57	0.00	4.25	4.07	3.60
3	0.86	4.25	0.00	5.33	7.31
4	6.12	4.07	5.33	0.00	3.70
5	7.85	3.60	7.31	3.70	0.00

Table 4.3. Percentage Relative Error in Range Data

Node	1	2	3	4	5
1	0.00	8.36	3.16	10.47	9.19
2	8.36	0.00	10.02	8.88	8.84
3	3.16	10.02	0.00	7.99	9.02
4	10.47	8.88	7.99	0.00	9.08
5	9.19	8.84	9.02	9.08	0.00

The effect of noise in the system is noticeable when nodes are placed in close proximity of each other. The algorithm seems to produce large errors in position estimates when nodes are placed within 2 meters of each other.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This thesis presented a hardware implementation of a sensor network that is able to localize itself and track a mobile target UAV/UGV in real time. The current implementation accommodates 8 nodes in total, with a minimum of 4 nodes forming the ground network while the rest can be used as tags on different target vehicles. The algorithm can be easily re-written to accommodate more nodes and the whole network can be made scalable. The results obtained in the previous chapter show that the localization algorithm does not perform very well for nodes that are separated by distances less than 2 meters.

A limitation of the sensor modules is that they can only be deployed with the ultrasonic sensors facing upward or towards other nodes. This restriction in module orientation makes initial deployment critical. One solution to this problem would be to attach multiple ultrasonic sensors to face different directions, so that no matter how they are deployed at least two or more sensors will not be occluded. This would increase the size or form factor of the PCB considerably. An alternative would be to use just the radios for localization and ranging, sacrificing distance measurement resolution for a smaller form factor. Because the sensor modules are custom developed by the author, any change in design can be done quickly with minimum overhead using off-the-shelf components if needed.

Currently the system is designed to be deployed and operated within a single room. In a multi-room environment it is possible that not all nodes are in communication range with each other, which means that data will have to be routed. A

communication protocol akin to TCP/IP would need to be developed to allow for effective routing. The nanoLOC radios have in-built signal collision detection and avoidance schemes as well as Forward Error Correction (FEC) and automatic packet retransmission capabilities, making the task easier. The network must be ad hoc in nature to accommodate newly discovered nodes or prune defective nodes at any time. If the target were to be controlled remotely using the sensor network, information would have to be optimally routed across the network to minimize lags between actual target movement and on screen display updates.

The sensor network can track a target's position but not its orientation. To be able to determine a UAV/UGV's orientation, an implementation similar to the CRICKET sensors will work [12], where an array of ultrasonic sensors placed at specific intervals from each other determine the phase difference of the incident signal to calculate a target's orientation. Another approach would be to attach multiple nodes at each extremity of the UAV/UGV and use their position information to determine target's heading and inclination.

Each sensor node is powered by a 9V battery. In order to prolong battery life, several changes can be made in software from reducing oscillator frequency to powering down peripherals during idle times. These measures will also extend the device life itself.

This system was designed primarily to be used for tracking multiple heterogeneous vehicles being developed at ARRI as part of an on-going research in formation control algorithms.

REFERENCES

[1] P. van Blyenburgh, "UAVs: An Overview," *Air and Space Europe*, vol. 1, no. 5-6, pp. 43–47, 1999.

[2] P. Rudol and P. Doherty, "Human Body Detection and Geolocalization for UAV Search and Rescue Missions Using Color and Thermal Imagery," in *Proceedings of the IEEE Aerospace Conference*, 2008.

[3] J. Bachrach and C. Taylor, "Localization in sensor networks," in *Handbook of Sensor Networks : Algorithms and Architecture*, vol. 1. Wiley,USA, 2005, pp. 277–304.

[4] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, October 2000. [Online]. Available: 0119

[5] A.-K. Chandra-Sekaran, P. Dheenathayalan, P. Weisser, C. Kunze, and W. Stork, "Empirical analysis and ranging using environment and mobility adaptive rssi filter for patient localization during disaster management," *Networking and Services, International conference on*, vol. 0, pp. 276–281, 2009.

[6] S. Y. Wong, J. G. Lim, S. V. Rao, and W. K. G. Seah, "Density-aware hop-count localization (dhl) in wireless sensor networks with variable density," in *IEEE Wireless Communications and Networking Conference (WCNC 2005)*, vol. 3, March 2005, pp. 1848–1853. [Online]. Available: http://dx.doi.org/10.1109/WCNC.2005.1424793

[7] S. Lee, E. Kim, C. Kim, and K. Kim, "Hop-count based localization using geometric constraints in wireless sensor networks," in *Communications, 2008. APCC 2008. 14th Asia-Pacific Conference on*, Oct. 2008, pp. 1–5.

[8] N. Kirchner and T. Furukawa, "Infrared localisation for indoor uavs," 2008.

[9] D. Niculescu and D. Lab, "Ad hoc positioning system (APS) using AoA," 2003, pp. 1734–1743.

[10] J. Li, X. Sun, P. Huang, and J. Pang, "Performance Analysis of Active Target Localization Using TDOA and FDOA Measurements in WSN," *Advanced Information Networking and Applications Workshops, International Conference on*, vol. 0, pp. 585–589, 2008.

[11] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," 2000, pp. 32–43.

[12] N. B. Priyantha, A. K. Miu, H. Balakrishnan, and S. Teller, "The Cricket Compass for Context-Aware Mobile Applications," in *7th ACM MOBICOM*, Rome, Italy, July 2001.

[13] F. Lewis, "Wireless Sensor Networks," in *Smart Environments: Technologies, Protocols, and Applications, ed. D. J. Cook and S. K. Das,*. New York, NY, USA: John Wiley, 2004.

[14] A. Tiwari, P. Ballal, and F. L. Lewis, "Energy-efficient wireless sensor network design and implementation for condition-based maintenance," *ACM Trans. Sen. Netw.*, vol. 3, no. 1, p. 1, 2007.

[15] Nanotron Technologies, "Real time location systems (rtls)," 2007.

[16] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Chapel Hill, NC, USA, Tech. Rep., 1995.

[17] D. Simon, "The Discrete-time Kalman Filter," in *Global Positioning Systems, Inertial Navigation, and Integration.* John Wiley and Sons, Inc., 2007, pp. 255–316.

[18] P. Dang, P. Ballal, F. L. Lewis, and D. O. Popa, "Real Time Relative and Absolute Dynamic Localization of Air Ground Wireless Sensor Networks," *J. Intell. Robotics Syst.*, vol. 51, no. 2, pp. 235–257, 2008.

[19] S. Qin-qin, H. Hong, F. Tao, and L. De-ren, "Using linear intersection for node location computation in wireless sensor networks."

[20] M. B. Middleton, P. E. Stingu, C. McMurrough, and F. L. Lewis, "Development of a testbed for rapid, structured design of intelligent controls and interfaces for indoor mobile robots."

[21] L. R. W. Mohinder S. Grewal and A. P. Andrews, "Kalman Filtering," in *Optimal State Estimation.* John Wiley and Sons, Inc., 2006, pp. 123–149.

BIOGRAPHICAL STATEMENT

Vishal Savio Coelho received his Bachelor of Engineering degree in Electronics and Tele-Communications from the University of Mumbai, India, in 2006. In the same year he started his Masters in Electrical Engineering at the University of Texas at Arlington. In the Fall of 2007 he joined Analog Devices, Inc. as an intern for the Product Definition and Applications team of the RFWS(Radio Frequency and Wireless) group. He returned to the university in 2008 to pursue research in the field of sensor networks under the supervision of Dr. Frank Lewis. He has been working as a Graduate Research Assistant at the Automation & Robotics Research Institute.

His current research interest is in the area of intelligent control and biomimetic robotics.