A tracking approach to localization and synchronization in mobile ad-hoc sensor networks

Pat Bidigare, Chris Kreucher and Ralph Conti

The General Dynamics Michigan Research and Development Center, Ypsilanti MI, USA

ABSTRACT

Self localization is a term used to describe the ability of a network to automatically determine the location of its nodes, given little or no external information. Self localization is an enabling technology for many future capabilities; specifically those that rely on a large number of sensors that self organize to form a coherent system. Most prior work in this area focuses on centralized computation with stationary nodes and synchronized clocks. We report on preliminary results for a setting that is more general in three ways. First, nodes in the network are moving. This implies the pair-wise distances between nodes are not fixed and therefore an iterative tracking procedure is needed to estimate the time varying node positions. Second, we do not assume synchronization between clocks on different nodes. In fact, we allow the clocks to have both an unknown offset and to be running at differing rates (i.e., a drift). Third, our method is decentralized, so there is no need for a single entity with access to all measurements. In this setup, each node in the network is responsible for estimating its state.

The method is based on repeated pair-wise communication between nodes. We focus on two types of observables in this paper. First, we use the time between when a message was sent from one node and when it was received by another node. In the case of synchronized clocks and stationary nodes, this observable provides information about the distance between the nodes. In the more general case with non-synchronized clocks, this observable is coupled to the clock offsets and drifts as well as the distance between nodes. Second, we use the Doppler stretch observed by the receiving node. In the case of synchronized clocks, this observable provides information about the line of sight velocity between the nodes. In the case of non-synchronized clocks, this observable is coupled to the clock drift as well as the line of sight velocity. We develop a sophisticated mathematical representation that allows all of these effects to be accounted for simultaneously.

We approach the problem from a Bayesian viewpoint, where measurements are accumulated over time and used to form a probability density on the state, conditioned on the measurements. What results is a recursive filtering (or tracking) algorithm that optimally merges the measurements. We show by simulation and illustrative data collections that our method provides an efficient decentralized method for determining the location of a collection of moving nodes.

Keywords: Self Localization, Self Organizing Coherent Systems

1. INTRODUCTION

Consider the four military/intelligence applications of network self-localization illustrated in Figure 1(L):

Radio Network Self-Localization A network of software definable radios uses RF ranging techniques to determine the locations of its radio nodes in an environment where GPS is unavailable (e.g., under dense foliage) or is denied. Absolute location is determined via a small number of nodes at surveyed locations.

Satellite Constellation Maintenance Using laser ranging between platforms and from platforms to surveyed ground sites, a constellation of satellites can obtain very accurate ephemeris (location) information.

Unmanned Underwater Vehicle (UUV) Networks A swarm of UUVs can determine the relative location of each vehicle in the cluster by transmitting acoustic signals between the vehicles. Additional stationary nodes (buoys) at surveyed locations allow absolute position determination of these vehicles.

Send correspondence to Christopher.Kreucher@gd-ais.com

Defense Transformation and Network-Centric Systems, edited by Raja Suresh,
Proc. of SPIE Vol. 6249, 62490I, (2006) · 0277-786X/06/$15 · doi: 10.1117/12.665618

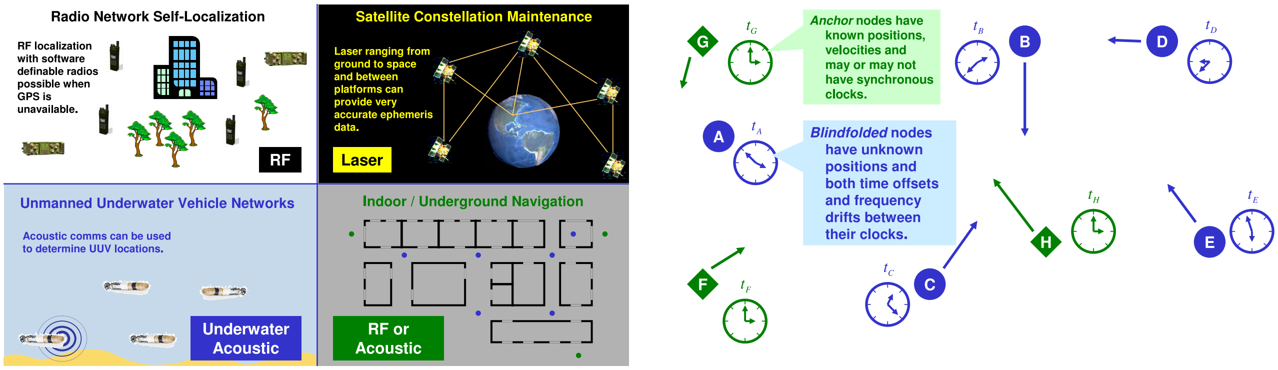

Radio Network Self-Localization
RF localization with software definable radios possible when GPS is unavailable.
RF
Satellite Constellation Maintenance
Laser ranging from ground to space and between platforms can provide very accurate ephemeris data.
Laser
Unmanned Underwater Vehicle Networks
Acoustic comms can be used to determine UUV locations.
Underwater Acoustic
Indoor / Underground Navigation
RF or Acoustic
G
t_G
Anchor nodes have known positions, velocities and may or may not have synchronous clocks.
t_B
B
D
t_D
A
t_A
Blindfolded nodes have unknown positions and both time offsets and frequency drifts between their clocks.
t_H
t_E
H
E
F
t_F
t_C
C

Figure 1. Left: Four applications of Network Self-Localization. Right: A network consists of moving nodes including *anchor* nodes, which have an external source of position/velocity and possibly time and *blindfolded* nodes, which have no external knowledge of their position/velocity and have asynchronous clocks.

Indoor / Underground Navigation A collection of small devices with RF or acoustic communication capabilities is randomly or selectively distributed in an indoor or underground environment. By transmitting signals between the nodes, and having 'way point' nodes at known locations, this network can determine its node's locations to facilitate underground navigation of mobile nodes.

In each of these examples, a network must localize its nodes while they are in motion. Additionally, in none of these examples is a central time reference necessarily present, so the network must deal with uncertain time offsets and drifts between the node clocks. In this paper, we report on our preliminary work in network self-localization as illustrated in Figure 1(R), for networks with moving nodes having no common time base.

Suppose there are a small number of anchor nodes, which have some external source of position, velocity and possibly timing. This external information could be from GPS, an INU, a WWV time broadcast, or a surveyed position. A sufficient number of such nodes is necessary if absolute position and time is to be estimated across the network. The majority of the nodes are blindfolded nodes, which have no external source of position or time information, because they are designed to be inexpensive, have severe power constraints or operate in an environment where no external positioning information is available. We assume that the nodes in our network are able to broadcast (using RF, acoustic, optical or other media) time-tagged messages to neighboring nodes.

A network node receiving a transmission can extract the embedded time tagged message. The time of arrival (TOA) of the message (in the receiver's time reference) along with its Doppler scale (DS) factor can also be measured. The TOA measurement depends on both the propagation time between the nodes and the offset between their clocks. Similarly the DS measurement depends on both the line-of-sight velocity between the nodes and the drift between their clocks. Propagation effects (depending on range and velocity) are reciprocal: the propagation time and its derivative from Alice to Bob is the same as from Bob to Alice. However, change of time base effects (clock offset and drift) are anti-reciprocal: the offset added to Alice's clock time to get Bob's is the negative of that needed to translate Bob's time into Alice's. This reciprocity distinction allows motion and synchronization parameters to be separately extracted from TOA / DS measurements when simultaneous, two-way measurements are made. In this paper we extend this idea by incorporating a distributed filtering approach that alleviates the need for simultaneous measurements and allows kinematic models to be employed.

There are several real phenomena that we will ignore in the paper development. These include multipath,[1] collisions between simultaneously talking nodes, and transmission sequencing. A practical moving network self localization scheme must address all of these.

The body of literature addressing network self-localization has been growing rapidly. Virtually all papers written so far have dealt with the case of stationary nodes. While most of these references use time of arrival measurements, there are other papers that consider angle of arrival and/or received signal strength[2,3] measurements. Most of the current literature deals with networks in which there is a common time reference, although concepts and architectures for dealing with asynchrony have been considered.[4] There are several good references that compute the Cramer Rao lower bounds for synchronous, stationary network self localization.[2, 5, 6] Various papers present practical algorithms for source localization. Centralized algorithms[5] (all processing occurs at a central node) obtain the tightest estimates, but may be impractical over many data rate limited networks. Distributed algorithms[3, 7–10] offer reasonable performance without the communications overhead.

Section 2 presents a quick introduction to Bayesian tracking and how this applies to our problem. We describe the state space our technique uses, along with the kinematic models and measurement (sensor) models. We present our particular tracker implementation. Section 3 presents particulars on how TOA and DS measurements are extracted from recorded signals and shows simulation results on how position uncertainty is reduced as tracking progresses. Section 4 presents the results of some acoustic experiments we performed with laptops and mobile robot bases. We show how clock offset and propagation time can be extracted from real data, as well as how line of sight velocity and clock drift can be distinguished. Many of the problems facing a practical implementation of acoustic self-localization are presented here, including sample synchronization, acoustic imperfections and multipath. Finally, we conclude with a description of the lessons learned in this research and the future work needed to make this technique practical.

2. BAYESIAN TRACKING

The goal of our work is to generate an on-line estimate of the state of a collection of moving platforms. We adopt a Bayesian filtering approach, which is a theoretically rigorous method for constructing a probability distribution on a state $\mathbf{x}$ by fusing measurements made over time with statistical models on state evolution and the measurement process. In this paradigm, rather than generating a point estimate of the state, one instead constructs a probability distribution on the state. This allows information to be accumulated over time through a series of measurements, and that detailed physical models of target motion and the sensor measurements can be used optimally. Of course, a point estimate can be computed from the probability density if desired.

To formalize, we wish to estimate a state $\mathbf{x}$ at time k, denoted $\mathbf{x}^k$. Let $\mathbf{z}^k$ denote the measurement (scalar, vector, or matrix) made at time k. Additionally, let $\mathbf{Z}^k$ denote the collection of measurements made up to and including time k, i.e., $\mathbf{Z}^k = \{\mathbf{z}^0, \mathbf{z}^1, \cdots, \mathbf{z}^{k-1}, \mathbf{z}^k\}$. Then the goal of Bayesian filtering is to construct the conditional probability density on the state $\mathbf{x}$ at time k, conditioned on all measurements up to and including time k, i.e., to construct $p(\mathbf{x}^k|\mathbf{Z}^k)$. If the density is estimated correctly, point estimates of the state can be easily generated. For example, the *Maximum A Posteriori* (MAP) estimate is $\hat{\mathbf{x}}^k_{MAP} = \arg\max_{\mathbf{x}^k} p(\mathbf{x}^k|\mathbf{Z}^k)$, and the minimum mean squared error (MMSE) estimate is $\hat{\mathbf{x}}^k_{MMSE} = \int_{\mathbf{x}^k} \mathbf{x}^k p(\mathbf{x}^k|\mathbf{Z}^k) d\mathbf{x}^k$.

Bayesian filtering is a recursive procedure for predicting and updating the conditional probability density by fusing measurements with statistical models of the state evolution and the measurement process. As such, it requires both a *model of state evolution* and a *model of sensor physics*. Both of these models are problem-specific and require a detailed knowledge of the problem physics. The model of state evolution, denoted $p(\mathbf{x}^{k+1}|\mathbf{x}^k)$, is a statistical description of how the state changes over time. This model may also be parameterized by a control input and written $p(\mathbf{x}^{k+1}|\mathbf{x}^k; \mathbf{u}^k)$. Similarly, the model of sensor physics, denoted $p(\mathbf{z}^k|\mathbf{x}^k)$ is a statistical description of how the measurements $\mathbf{z}^k$ are related to the state being estimated. We discuss the specification of each of these models in our problem in the following subsections.

With these models in hand, Bayesian filtering becomes a two-step recursion. First, the posterior at time k, $p(\mathbf{x}^k|\mathbf{Z}^k)$ is predicted forward in time using the model of state evolution to generate the prior $p(\mathbf{x}^{k+1}|\mathbf{Z}^k)$, via

$$p(\mathbf{x}^{k+1}|\mathbf{Z}^k) = \int_{\mathbf{x}^k} p(\mathbf{x}^k|\mathbf{Z}^k) p(\mathbf{x}^{k+1}|\mathbf{x}^k) d\mathbf{x}^k \quad . \tag{1}$$

Second, the prior $p(\mathbf{x}^{k+1}|\mathbf{Z}^k)$ is updated using the most recent measurements via Bayes' rule as

$$p(\mathbf{x}^{k+1}|\mathbf{Z}^{k+1}) = \frac{p(\mathbf{x}^{k+1}|\mathbf{Z}^k) p(\mathbf{z}^{k+1}|\mathbf{x}^{k+1})}{p(\mathbf{z}^{k+1}|\mathbf{Z}^k)} \quad . \tag{2}$$

Using this two-step procedure, one can synthesize measurements and the statistical models of the problem to generate estimates of the state parameters at any time. In the following subsections, we discuss the modeling choices involved in specializing this method to the self localization problem.

2.1. Definition of the State Vector

The state, denoted $\mathbf{x}$, is a vector entity describing the parameters the algorithm is to estimate. In the present application, we are interested in estimating the kinematic state (position and velocity) and clock parameters of a collection of moving platforms. We assume motion in two dimensions for purposes of discussion, but the approach easily generalizes to three dimensions.

There are two competing choices for the state model that we discuss here. The most straightforward state definition is to specify all of the things to be estimated. Define by δt and $\dot{\delta t}$ the clock offset and drift from global, respectively. For our purposes, the algorithm is to estimate the kinematic state and the clock parameters of each of the N sensors. Therefore, one choice of the state is

$$\mathbf{x} = [x_1, \dot{x}_1, y_1, \dot{y}_1, \delta t_1, \dot{\delta t}_1, \cdots, x_N, \dot{x}_N, y_N, \dot{y}_N, \delta t_N, \dot{\delta t}_N] \ . \tag{3}$$

This choice of state model has several benefits. First, the Bayesian recursions outlined above are directly applicable, since all of the parameters of interest are captured by the state. Second, the inherent couplings and correlations between the platforms are explicitly captured. However, this definition of state carries several drawbacks. First, it is a very high dimensional state ($6N$) and hence requires sophisticated algorithms for implementation and significant computational resources at each platform. Second, this approach does not lend itself to decentralized estimation. If each platform is to estimate this state accurately, each platform must be privy to all measurements in the network, even those that only involve other platforms. Third, this method does not naturally lend itself to sensors arriving and leaving (which would change the state space dimension for a state defined in this manner).

A second choice for the state, and one that we adopt here, is to simply define the state as

$$\mathbf{x} = [x, \dot{x}, y, \dot{y}, \delta t, \dot{\delta t}] \ , \tag{4}$$

and have each node estimate its own state. Therefore, each node tracks its own 6-dimensional state rather than tracking the $6N$-dimensional state of the entire network. This approach has several benefits. First, the approach is naturally decentralized (each node is only estimating things about itself) and lends itself to nodes coming online and going offline while running. Second, the computational burden is low as each node is only doing single-target tracking. There are, of course, drawbacks to this sparse representation. As each node is estimating only part of the global state vector, the approach is inherently deficient in estimating correlations and couplings between the states of different nodes. Furthermore, as we will see in later subsections, to compute the measurement update a node must know some information about other nodes. This requires additional communication and introduces additional imprecisions in the algorithm, including data incest.

2.2. Model of State Evolution

Once the target state is defined, the next step is to formulate a model of state evolution, $p(\mathbf{x}^{k+1}|\mathbf{x}^k)$. This is a statistical model that describes how the state of the system evolves with time. Many models are plausible (even multiple models), and in general a model is formulated after detailed study of the target physics.

The simplest model is one where the sensor motion is completely deterministic, i.e., its position at time k completely defines its position at time $k+1$. This may be the case in situations where the nodes move under their own power in a completely known and predictable manner. For example, if a node decides to move $1m$ to the left, the action is executed flawlessly and the node in fact moves exactly $1m$ to the left. In this case, the statistical model would simply be

$$p(\mathbf{x}^{k+1}|\mathbf{x}^k; \mathbf{u}^k) = \delta(\mathbf{x}^{k+1} - f(\mathbf{x}^k, \mathbf{u}^k)) \ , \tag{5}$$

where f is a deterministic function of the previous state and control input. This approach requires the unrealistic assumptions that control inputs are known exactly, executed exactly, and no outside force influences the motion.

For these reasons, we use a more general model of target evolution that incorporates an error term to model motion imperfections. In this approach, it is assumed that the nodes move according to the nearly constant velocity (NCV) model.[11] Denote the time between updates of the filter (i.e., the time between discrete time tick k and discrete time tick $k+1$) by T. Then the model is completely specified by

$$\mathbf{x}^{k+1} = F\mathbf{x}^k + Q\mathbf{v}^k. \tag{6}$$

where $\mathbf{v}^k$ is 0-mean Gaussian noise, and

$$F = \begin{pmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad Q = q \begin{pmatrix} T^3/3 & T^2/2 & 0 & 0 & 0 & 0 \\ T^2/2 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & T^3/3 & T^2/2 & 0 & 0 \\ 0 & 0 & T^2/2 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & q_{\delta\delta} & q_{\delta\dot{\delta}} \\ 0 & 0 & 0 & 0 & q_{\dot{\delta}\delta} & q_{\dot{\delta}\dot{\delta}} \end{pmatrix} . \tag{7}$$

2.3. Sensor Model

Definition of the sensor model requires a detailed physical understanding of how the measurements relate to the states under estimation. In our application, measurements are generated in the following manner. When Alice chooses to communicate with other nodes, she does so by sending a predefined signal. It furthermore encodes the time of transmission in its local time frame, C_A^{TX} and statistics describing its estimate of its own state, $p(\mathbf{x}_A^k|\mathbf{Z}^k)$. When Bob receives the communication at time C_B^{RX}, he records the delay from transmission to reception (which includes the actual delay and effects of clock mis-synchronization from Alice to Bob). Furthermore, Bob also records the Doppler stretch $\dot{d}$ as seen in its local time frame. Therefore, the measurements Bob receives are given by $\mathbf{z}^k = [d, \dot{d}]$, with (known) parameters $[C_A^{TX}, p(\mathbf{x}_A^k|\mathbf{Z}^k)]$.

To compute the measurement update, we then need to specify the sensor model

$$p(\mathbf{z}^k|\mathbf{x}_B^k) = p(d|\mathbf{x}_B^k)p(\dot{d}|\mathbf{x}_B^k) \ , \tag{8}$$

where for the purposes of this paper, we assume the two measurements are independent.

It can be shown that in the absence of noise, these measurements are related to the state parameters in the following manner:

$$d = t^{TX} - \frac{\dot{t}^{TX}}{1-\dot{\tau}^{TX}}\left(\dot{\tau}^{TX}(C_A^{TX} - t^{TX}) + \tau^{TX} - C_B^{RX} - \frac{t^{RX}}{\dot{t}^{RX}}\right) \tag{9}$$

$$\dot{d} = \frac{\dot{t}^{RX}(1-\dot{\tau}^{TX})}{\dot{t}^{TX}} \tag{10}$$

where t^{TX} is the transmission time in global clock, $\dot{t}^{TX}$ is Alice's clock rate at transmission, τ^{TX} is the propagation time between Alice and Bob at transmission (which is proportional to the distance between the two), and $\dot{\tau}^{TX}$ is the rate of change of propagation time between Alice and Bob at transmission (which is proportional to the line of sight velocity between the two). All of these parameters are known probabilistically from Alice's p.d.f. Finally, $\dot{t}^{RX}$ is Bob's clock rate at reception.

Restricting our state space to include only the node's own position and time introduces a technical difficulty in updating the state when a new measurement arrives. The time-of-arrival and Doppler scale depend on both the states of the transmitter and the receiver. To deal with this, we marginalize the measurement p.d.f. over the transmitters state. For example, in order to compute $p(\dot{d}|\mathbf{x}_B^k)$, we need to know $\dot{t}^{RX}$, $\dot{\tau}^{TX}$ and $\dot{t}^{TX}$. Since we are computing the likelihood for a particular $\mathbf{x}_B^{RX}$, we know $\dot{t}^{RX}$. The other unknowns, $\dot{\tau}^{TX}$ and $\dot{t}^{TX}$ are known probabilistically from the information that Alice sends. Therefore, we can compute

$$p(\dot{d}|\mathbf{x}_B^{RX}) = \int p(\dot{d}|\mathbf{x}_B^{RX}|\mathbf{x}_A^{TX})p(\mathbf{x}_A^{TX})d\mathbf{x}_A^{TX} \ . \tag{11}$$

The remaining difficulty is how to compute the likelihood $p(\dot{d}|\mathbf{x}_B^{RX}|\mathbf{x}_A^{TX})$. That is – given information about Bob at reception and Alice at transmission, we need to compute the probability of a measurement. Equation

(10), which describes how $\dot{d}$ is related to the states of Bob and Alice, relies on $\dot{\tau}^{TX}$, which is implicitly defined by $\mathbf{x}_A^{TX}$ and $\mathbf{x}_B^{TX}$, which are not known. Getting at this would require $\mathbf{x}_B^{RX}$ to $\mathbf{x}_B^{TX}$, which requires us to know, in Bob's time base, when the pulse is sent. We instead make the assumption that Bob's parameters move slowly during the time that the pulse is in flight. Explicitly, we assume

$$p(\dot{d}|\mathbf{x}_B^{RX}|\mathbf{x}_A^{TX}) \approx p(\dot{d}|\mathbf{x}_B^{TX}|\mathbf{x}_A^{TX}) \ . \tag{12}$$

Finally, a statistical model on $\dot{d}$ is required. A reasonable assumption is probably that its mean is described by eq. (10) and it has some standard deviation about that mean, called σ_1, i.e., the model is

$$\dot{d} \sim N\left(\frac{\dot{t}^{RX}(1-\dot{\tau}^{TX})}{\dot{t}^{TX}}, \sigma_1\right) \ . \tag{13}$$

This allows computation of eq. (11). A similar procedure holds for d.

2.4. Implementation

In this section, we discuss the implementation of the tracker. The most common implementational device is the ubiquitous Kalman Filter. There are a long list of reasons to adopt the Kalman Filter including its inherent tractability, robustness and the fact that it is optimal for a certain class of problems. However, the Kalman Filter requires several restrictive assumptions: the state PDF is Gaussian, the target motion model is linear, and the measurement model is linear. The first and third of these assumptions are violated in our application. Therefore, a more general approach is appropriate.

Several modifications of the Kalman Filter are popular, including the Extended Kalman Filter, the Unscented Kalman Filter, and the Gaussian sum Filter. They all, however, continue to make the assumption that the state PDF is well represented by a Gaussian – which is not the case in our application.

For these reasons, we adopt a particle filter approximation of the relevant probability densities. Particle filtering is a more general approach to Bayesian filtering that admits non-Gaussian densities and non-linear measurement to state coupling. Particle filtering is a method of approximately solving the prediction and update equations by simulation[12],[13] where samples from the target density are used to represent the density and are propagated through time. As the number of particles tends towards infinity, the approximation converges to the true posterior being approximated.[14]

In a particle filter, the density of interest, $p(\mathbf{x}|\mathbf{Z})$, is approximated by a set of weighted samples (particles):

$$p(\mathbf{x}|\mathbf{Z}) \approx \sum_{p=1}^{N_{part}} w_p \delta_D(\mathbf{x}-\mathbf{x}_p) \ , \tag{14}$$

where δ_D represents the usual Dirac impulse, $\mathbf{x}_p$ represent the particle locations, and w_p are the particle weights. The model update and the measurement update are simulated by the following three step recursion.

First, the particle locations at time k are generated using the particle locations at time $k-1$ and the current measurements by sampling from an importance density, denoted $q(\mathbf{x}^{k+1}|\mathbf{x}^k, \mathbf{z}^{k+1})$. The design of the importance density is a well studied area,[14] since the choice of the importance density has a dramatic effect on the efficiency of the algorithm. It is known that the optimal importance density (OID) is given by $p(\mathbf{x}^{k+1}|\mathbf{x}^k, \mathbf{z}^{k+1})$, but this density is typically prohibitively difficult to sample from. In practice, the importance density is often chosen just to be the kinematic prior $p(\mathbf{x}^{k+1}|\mathbf{x}^k)$. However, more sophisticated choices of importance density lead to better results for a fixed number of particles.

Second, particle weights are updated according to the weight equation (15), which involves the likelihood, the kinematic model, and the importance density.[12]

$$w_p^{k+1} = w_p^k \frac{p(\mathbf{z}^{k+1}|\mathbf{x}_p^{k+1})p(\mathbf{x}_p^{k+1}|\mathbf{x}_p^k)}{q(\mathbf{x}_p^{k+1}|\mathbf{x}_p^k, \mathbf{z}^{k+1})} \ . \tag{15}$$

Finally, a resampling step is used to combat particle degeneracy. Without resampling, the variance of the particle weights increases with time, yielding a single particle with all the weight after a small number of iterations.[14] Resampling may be done on a fixed schedule or based on the weight variance.

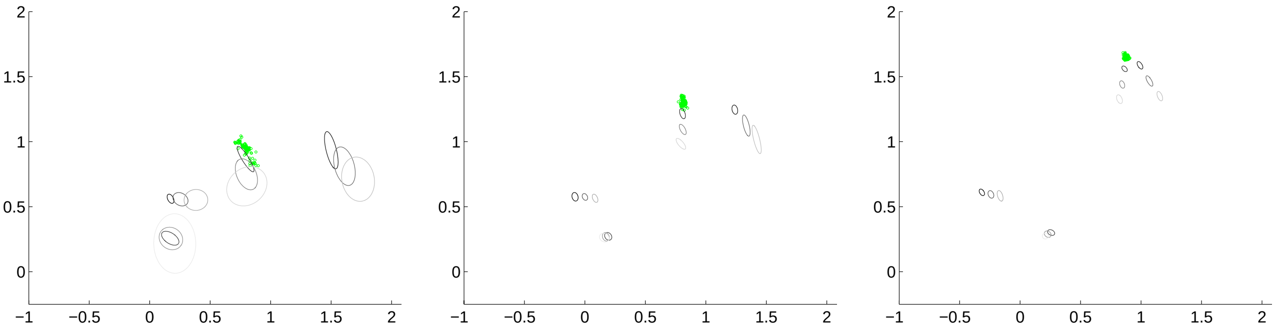


Figure 2. Three snapshots of information known to simulation node 2. 100 particles model each client's own knowledge of their position and uncertainty. Black uncertainty ellipses give latest knowledge on all nodes. Faded ellipses are from previous transmissions with lightness proportional to age.

3. SIMULATION

Before attempting a physical data collection, two kinds of simulations were run. In the first, a single MatLab program modeled all nodes in the sensor array and communications between them. In the second simulation, each node was run on a separate computer and the signal propagation channel was modeled by an additional master computer. Since the results were similar, only the results of the latter simulation are reported here.

Several simplifying assumptions are employed for these simulations: all node clocks are synchronized (no offset or drift from the global clock); point estimates of a node's parameters are used rather than its entire PDF when marginalizing; and a good initial estimate of each node position is assumed. These assumptions allow a more tractable implementation, to be relaxed in further research.

3.1. Simulation Implementation

The master computer that models the channel synchronizes communications with the clients and distributes exact knowledge of the clients' state. It also generates the signal with appropriate delay and Doppler stretch and adds channel noise. Each of the four client nodes, acting independently, prompt the master to send the waveform with appropriate delay and stretch to each of the other clients, along with the transmitting client's current estimate of its own position and velocity (X_{Tx}, v_{Tx}) and their uncertainties $(\sigma_{XTx}, \sigma_{vTx})$. Each receiver then uses a 2-stage coarse-to-fine CAF to determine $(\|X_{Tx} - X_{Rx}\|, \|v_{Tx} - v_{Rx}\|)$ and their measurement uncertainties to update their own (X_{Rx}, v_{Rx}) and $(\sigma_{XTx}, \sigma_{vTx})$. The transmitted waveform was a linear (V) chirp. This was chosen, because it gives moderately good noise immunity,[15] while remaining analytic for easy Doppler state estimation.

3.2. Simulation Results

The performance of the simulation is illustrated in Figure 2. These snapshots represent localization information available to node 2 early, middle, and late in the simulation. The grey dots represent the particles in the filter tracking node 2's movement. Node 2's knowledge of the tracked motion of all nodes is shown in the age faded uncertainty ellipses, derived from node 2's own filtering or transmissions from the other nodes. Twelve ellipses are shown (3 for each node), recording the position of one node at each of the previous 12 time steps.

Note the shrinking of the uncertainty ellipses as successive measurements are made. Not shown are the actual locations of each node, however, at the instants pictured, all are consistent with the uncertainty ellipses. Since no anchor nodes are included in this simulation, the ellipses eventually wander off the true positions, however, the relative positions of the nodes remain accurate, *i.e.* there can be translational and rotational drift only for the overall configuration of the nodes.

4. EXPERIMENTAL DATA COLLECTS

To illustrate the phenomena discussed in this paper, we performed a series of experimental data collects. In each data collect, platforms communicate by emitting and receiving acoustic waveforms. The waveform emitted by one platform is received by others, along with ancillary information regarding the transmission time. As discussed earlier, the measured time of arrival and Doppler stretch are coupled to both the relative clock parameters and the relative kinematic parameters of the sender and receiver.

This section reports on the findings of two of these data collects. We first report on some of the unmodeled error sources present in the data collects in subsections 4.1 through 4.3. These include the difficulty of accurately time stamping data, multipath effects, and the imperfections in the acoustic hardware. We then turn to the actual data collects. Both collects were done using standard Dell$^{\text{TM}}$ laptops with stock internal speakers and microphones. The first collect involves three stationary platforms and illustrates clock offset and drift across the platforms. The second collect involves a mix of stationary and moving platforms and illustrates the coupling of clock offset and drift to relative position and velocity.

4.1. Sample Synchronization

Sample synchronization refers to the ability of the hardware to accurately transmit a waveform at precisely the requested time as well as the ability to exactly time tag a received waveform. Early evidence suggested that direct control of the sound cards, specifically the transmit and receiver buffers, is required.

Through low level manipulations of the hardware, it is possible in general to receive accurate time tags. However, it was found during the data collection that the hardware often suffered from “glitches” wherein samples were dropped or mistimed. This is most likely due to a combination of factors, including buffer limits. In our work, we simply ignore those communications where hardware glitches occur.

4.2. Multipath

A second practical consideration that is not modeled explicitly in the tracker formulation is multipath. In a closed environment with walls, ceilings, and floors, a platform often receives both a direct path and reflection path copies of a transmitted waveform. Broadly speaking, we characterize the multipath as “microscopic” or “macroscopic”. Macroscopic multipath describes indirect path receptions that are a result of macroscopic features in the environment – e.g., walls, ceilings, and floors. This multipath is often offset from the direct path by a large distance (several meters), suggesting a simple gating procedure to eliminate the problem. However since it is possible that the indirect path reception is stronger than the direct path, and the platforms are moving simple range gating may not always be possible. Microscopic multipath describes indirect path receptions that are a result of small scale features, such as the electronics at transmission or reception. Figure 3 shows the cross correlation between a (known) transmitted waveform and a received waveform. This illustrates a reception that has been subject to both macroscopic and microscopic multipath.

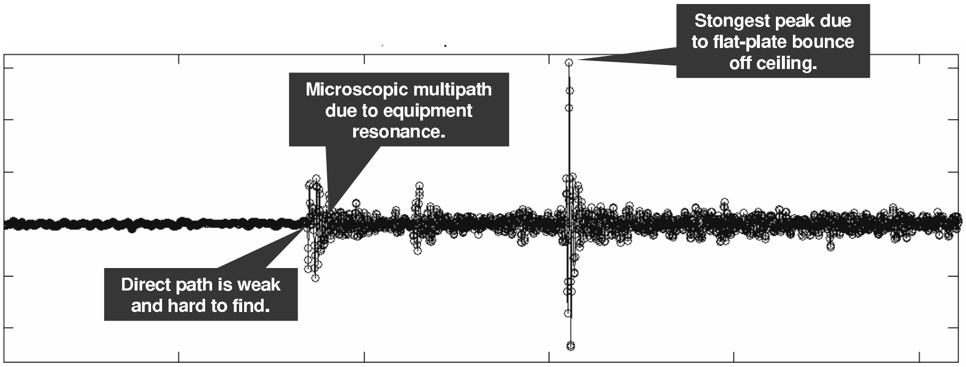


Figure 3. An example of multipath in the cross correlation function. The direct path is weaker than then bounce of the ceiling and microscopic multipath due to equipment resonance is present.

4.3. Acoustic Imperfections

Another practical consideration that is not modeled explicitly in the tracker formulation is the acoustic imperfections of the transmitter and receiver. Based on theoretical considerations, it was decided that acoustic communication would be done using a V-chirp waveform. However, as Figure 4 illustrates, with stock acoustic hardware the received waveform has quite a bit of distortion from the ideal waveform.

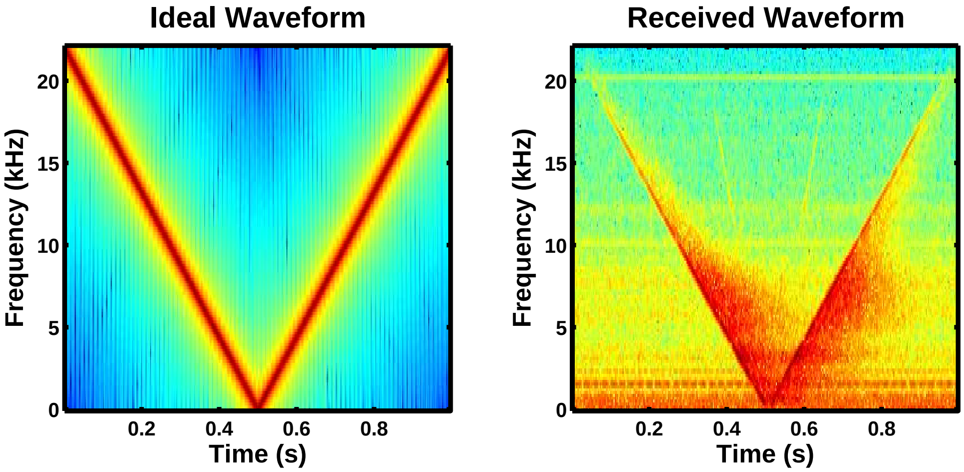


Figure 4. Spectograms illustrating the acoustic imperfections present in the experiment. Left: The spectogram of the ideal waveform that was nominally transmitted by Alice. Right: The spectogram of the waveform actually received by Bob. The imperfections are a combination of the non-ideal transmission from Alice and the non-ideal reception at Bob.

The collection environment was designed to minimize acoustic imperfections. Microphone and speaker gains were adjusted to avoid saturation while maintaining an adequate signal-to-noise ratio. Controllable background noise was minimized during the collect. Despite these efforts, the transmission is far from ideal. It includes non-flat frequency response as higher frequencies are subject to more attenuation than lower frequencies. Additionally, residual higher harmonic response is present. Also, there is quite a bit of spreading at low to mid frequencies. Finally, the collected data includes both low frequency and high frequency noise sources that omnipresent.

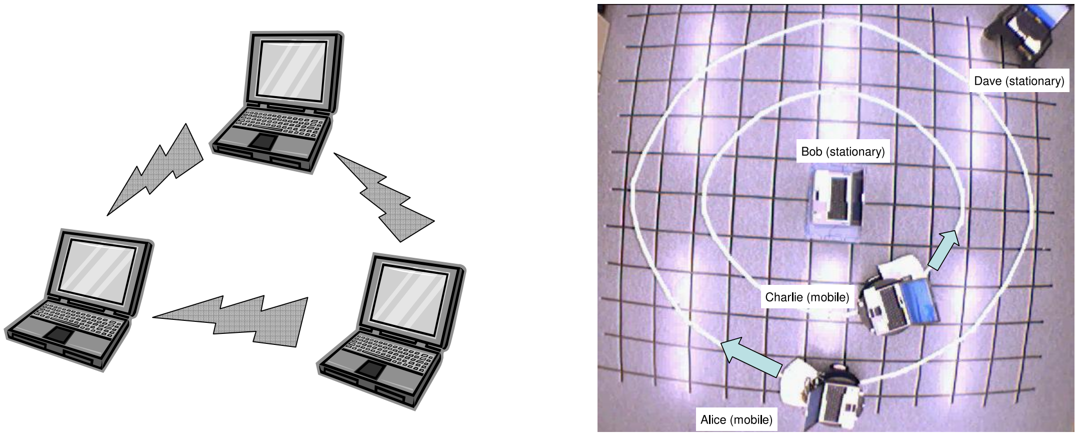


Figure 5. The two experimental data collects. Left: Three stationary computers performed repeated acoustic communications to illustrate clock offset and drift. Right: Data from a collection using a mix of mobile and stationary platforms illustrates the complex coupling of relative distance, motion, clock offset, and drift in the observables.

4.4. Stationary Platform Data Collection

The first experimental data collect involved stationary computers that had clocks that were unsynchronized. Figure 5(L) illustrates the configuration.

Each computer transmitted a distinct chirp pulse which was collected by the other computers. This resulted in a total of 6 pairwise communications. Since there are several experimental imperfections, this process was repeated 10 times, with each transmission spaced $5s$ apart. In this collection, we looked at the relative clock offset and drifts across the three computers. Let $\Delta t_{B,A} = C_B^{TX} - C_A^{RX}$, i.e., the transmission time as measured in Bob's clock minus the reception time as measured in Alice's clock. Define $\Delta t_{A,B}$ similarly. If the clocks were synchronized, the time between transmission and reception would be proportional to the distance between the two platforms and reciprocal, i.e., $\Delta t_{A,B} = \Delta t_{B,A} = D/c$, where D is the distance between nodes and c is the speed of sound. If the clocks were mis-synchronized by an offset only, the offset would effect the timing measurement so $\Delta t_{A,B} = D/c + O_{A,B}$ and $\Delta t_{A,B} = D/c - O_{A,B}$. If the clocks had both an offset and a drift, then this number would further change with time (over the 10 pulses) at a rate given by the relative clock drift.

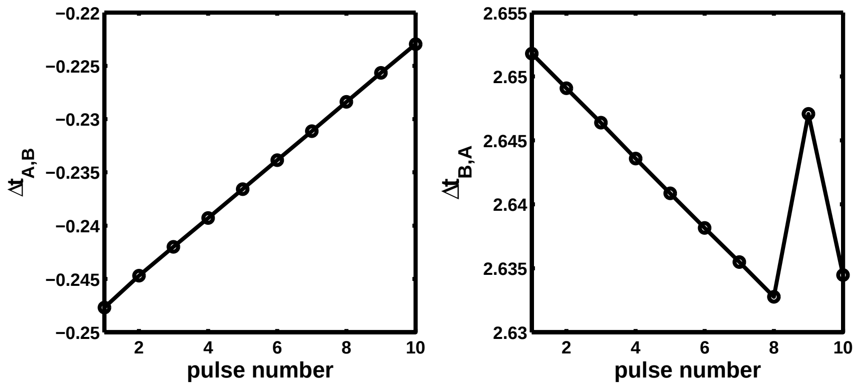


Figure 6. Plots of $\Delta t_{A,B}$ and $\Delta t_{B,A}$ for 10 pulses. The fact that $\Delta t_{A,B}$ and $\Delta t_{B,A}$ are not equal in magnitude indicates there is a clock offset. Furthermore, the fact that each is changing over the experiment indicates there is a clock drift.

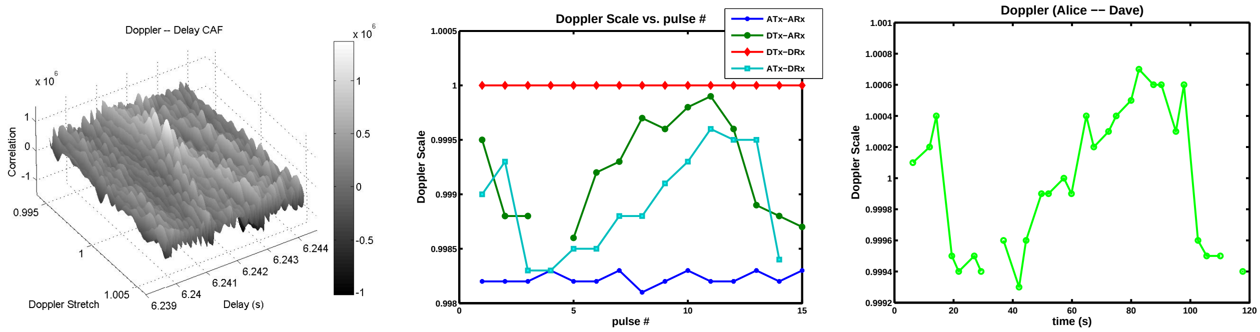
Figure 6 shows the measured values of $\Delta t_{B,A}$ and $\Delta t_{A,B}$. There is both an offset and a drift between the two computers. There are two pulses for which either the transmission time or reception time were inaccurately recorded and hence ignored in this discussion. Simple line fitting shows that the relative clock rates are equal and opposite (as expected) and that the clocks have rates that differ by approximately 1 part in 2000.

4.5. Moving Platform Data Collection

The second experiment was made up of a mix of stationary and moving platforms as captured in Figure 5(R). Acoustic transmissions were made in the same manner as described in the stationary experiment, but now the measured parameters are affected by both clock imperfections as well as the time varying platform positions.

Clock drift and relative velocity both affect the measured Doppler stretch of the transmitted signals. Clock offset and relative position both affect the measured time of arrival of the transmitted signals. These effects can be disentangled by measuring the Doppler stretch and time delay from two-way communications. An example of a two-dimensional (delay and Doppler stretch) cross ambiguity function (CAF) is shown in figure 4.5(a). The highest peak in the 2-D CAF marks the delay and Doppler stretch for a given received signal. The presence of other high peaks in the CAF at the "wrong" delay and Doppler stretch can be attributed to resonances in the acoustic hardware. In the presence of noise, the wrong peaks are occasionally chosen.

In the mobile experiment, Alice moves one and a quarter turns around an oval path, and Dave is stationary just outside Alice's oval. Alice, Bob, Charlie, and Dave take turns transmitting the linear chirp (each transmitter having a distinctive pulse length). Figure 4.5(b) details the Doppler stretch for pulses received by Alice and Dave during the data collect. The platform that transmitted the signal is denoted via Tx and the platform the received by Rx. Note that signals transmitted and received by the same node should exhibit no Doppler shift, as shown in the data for Dave(Tx)→Dave(Rx). However, on Alice the clock rate for transmit differs by 0.0018 from that for receive. Apparently the sound card on Alice (Dell latitude D600) has separate unsynchronized clocks for send and receive, while Dave's (Dell inspiron 8200) does not.



(a) Typical Doppler/delay CAF

(b) Doppler scale measurements for Alice (moving) and Dave (stationary).

(c) Corrected Doppler scale measurements between Alice and Dave.

Figure 7. Results of the moving platform experimental data collect

The Alice→Dave and Dave→Alice data reflect the same relative motion between Alice and Dave, and can be used to calibrate Alice's transmit and receive clocks to Dave's clock. After proper synchronization, the Alice–Dave data can be combined into a single curve (figure 4.5(c)) that measures the relative Doppler shift between Alice and Dave. This curve matches well with the Doppler shifts deduced from the over-head video of the data collect (in which pulse timing is clearly audible). In particular, when Alice's motion is perpendicular to the line-of-sight between Alice and Dave, no Doppler stretch is detected. The maximum velocity component observed was $v_{max} = 0.0006 \times 345\text{m/s} = 0.2\text{m/s}$. Thus, both clock drift and true Doppler shifts can be disentangled.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented the results of preliminary work on a distributed tracking based approach for localizing moving nodes in a network in which there is no common source of time synchronization. Our initial simulations show that this technique can iteratively reduce the uncertainty in network node position. Further, acoustic experiments using laptop computers on mobile robotic bases show that distance and relative velocity can be extracted from time of arrival and Doppler stretch measurements.

While this preliminary work forms a foundation for a distributed tracking based self-localization scheme, there are several technical issues that must be addressed before this scheme can be made practical.

Multipath: Our experiments show that both the macroscopic multipath from reflections off nearby objects and the microscopic multipath from equipment resonances are severe and unavoidable. Often multipath components can be significantly stronger than the direct path signal received. Various approaches to multipath include estimating 'earliest' return and using a statistical model to predict the time of the direct path return from observed multipath echoes.[1] Another approach is to incorporate multipath by modeling the various components and using a multiple hypothesis tracker to sort out the most likely direct path component based on consistency with other measurements taken.

Time Tagging: When accurate times of arrival are to be measured, nodes must be able to accurately tag their message transmission and reception times. Very often, uncontrollable delays can be introduced in the MAC layer of a transmission or because of processor interrupts in a non real-time operating system. These effects introduce unknown biases in the measurements that ultimately limit the localization fidelity. Practical systems may require specialized protocols along the OSI stack to insure measurement accuracy.

Uncertain Measurement Times: Conventional tracking problems allow uncertainty in the quantities being measured, but rarely deal with the case where the *time* that a measurement was taken is not known precisely. In our application, two-way measurements are needed to resolve propagation effects from synchronization effects. The time at which Alice receives a message from Bob is well-defined only in Alice's time frame, and so in Bob's time base, there is uncertainty about when this measurement was taken. A proper treatment of this would take into account the uncertainty in when a measurement was taken.

Networking Issues: We haven't addressed issues associated with ad-hoc wireless network implementation such as transmission protocols. This implementation will strongly affect self-localization accuracy.

ACKNOWLEDGMENTS

The authors wish to thank John Wegrzyn, Jeff Pursell, and Leah Shackman for their assistance in this work.

REFERENCES

1. J. Y. Lee and R. A. Scholtz, "Ranging in a dense multipath environment using an uwb radio link," *IEEE J. Selected Areas in Communications* **20**, pp. 1677–1683, December 2002.
2. E. A. Neal Patwari, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing* **51**, pp. 2137–2148, August 2003.
3. J. A. Costa, N. Patwari, and A. O. Hero, "Distributed weighted multidimensional scaling for node localization in sensor networks." To appear in ACM Journal on Sensor Networks.
4. D. D. McCrady et. al., "Mobile ranging using low-accuracy clocks," *IEEE Trans on Microwave Theory and Techniques* **48**, pp. 951–957, June 2000.
5. E. G. Larsson, "Cramer-Rao bound analysis of distributed positioning in sensor networks," *IEEE Signal Processing Letters* **11**, pp. 334–337, March 2004.
6. E. A. N. Patwari, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine* **22**, pp. 54–69, July 2005.
7. D. Niculescu and B. Nath, "Ad hoc positioning system (aps)," in *Proc. Global Telecommunications Conference*, **5**, pp. 2926–2931, November 2001.
8. C. Savarese, J. M. Rabaey, and J. Beutel, "Location in distributed ad-hoc wireless sensor networks," in *Proc. ICASSP*, **4**, pp. 2037–2040, May 2001.
9. E. A. A. Savvides, "An analysis of error inducing parameters in multihop sensor node localization," *IEEE Trans. Mobile Computing* **4**, pp. 567–577, Nov-Dec 2005.
10. K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks:a quantitative comparison," *Computer Networks* **43**, pp. 499–518, 2003.
11. Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, 1988.
12. M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Processing* **50**, pp. 174–188, Feb 2002.
13. N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "A novel approach to non-linear and non-Gaussian Bayesian state estimation," *IEE Proceedings on Radar and Signal Processing* **140**, pp. 107–113, 1993.
14. A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001.
15. Q. Jin, K. Maxwong, and Z.-Q. Luo, "The estimation of time delay and doppler stretch of wideband signals," *IEEE Transactions on Signal Processing* , pp. 904–916, April 1995.