inTrack: High Precision Tracking of Mobile Sensor Nodes

Branislav Kusý[1], György Balogh[1], János Sallai[1], Ákos Lédeczi[1], and Miklós Maróti[2]

[1] Institute for Software Integrated Systems,
Vanderbilt University, Nashville, TN, USA
akos.ledeczi@vanderbilt.edu
http://www.isis.vanderbilt.edu
[2] Department of Mathematics,
University of Szeged, Hungary
mmaroti@gmail.com

Abstract. Radio-interferometric ranging is a novel technique that allows for fine-grained node localization in networks of inexpensive COTS nodes. In this paper, we show that the approach can also be applied to precision tracking of mobile sensor nodes. We introduce inTrack, a cooperative tracking system based on radio-interferometry that features high accuracy, long range and low-power operation. The system utilizes a set of nodes placed at known locations to track a mobile sensor. We analyze how target speed and measurement errors affect the accuracy of the computed locations. To demonstrate the feasibility of our approach, we describe our prototype implementation using Berkeley motes. We evaluate the system using data from both simulations and field tests.

1 Introduction

Node localization is an important service for sensor network applications because in order to correlate the observations of a physical phenomenon made by different sensor nodes, their precise locations need to be known. When some of the sensors are mobile, they need to be tracked continuously. Alternatively, one can envision applications where the whole purpose of the sensor network is to track some objects, for instance, vehicles, people or animals, and a node is mounted on each object for this purpose. The location data may be used locally for navigation, for example, but it is typically transmitted to a remote computer and used, for instance, to display the position of the object on a map in real-time. GPS navigation systems and tracking services are examples of such applications outside of the sensor network realm. However, when low-cost, low-power and/or high precision are required and the area of interest is relatively constrained in size, a wireless sensor network can provide an ideal platform.

In this work, we concentrate on cooperative tracking, in which the tracked object works together with the tracking system to determine its location. We envision a fixed set of resource constrained wireless devices deployed at known locations forming the infrastructure of the tracking service. Their function is the same as that of the GPS satellites: they act as RF sources for the purpose of locating objects relative to their known positions.

Low-power, low-cost devices typically offer only limited computational power, memory and radio range. Despite these limitations, it has been shown that it is possible to measure highly accurate ranges over large distances [1]. The radio-interferometric positioning system presented in [1] was further improved in [2] to achieve a demonstrated localization error of less than 10 cm at a remarkably low node density of 650 nodes/km^2.

Previous work on radio-interferometric positioning in sensor networks focused exclusively on networks with stationary nodes. The single limiting factor making tracking a more challenging problem than node localization is the non-zero time required to measure the ranges, constraining both the real-time availability of the location estimates and the maximum allowable speed of the tracked object. On one hand, localization accuracy depends on the amount of ranging data available. Increasing the number of ranging measurements, however, limits the refresh rate. On the other hand, if the speed of the tracked object is high, the range changes significantly during the measurement and inconsistent ranging data is obtained. Radio-interferometric ranging is especially prone to these errors, because a single range measurement takes anywhere between 100 ms and 1 sec, depending on the required ranging accuracy and also on the capabilities of the hardware platform.

The objectives for a practical and effective tracking system are:

Accuracy: location errors are less than a meter error on average.
Large scale: the node density is less than 1000 nodes/km^2.
Real-time tracking: refresh rate and latency are adequate for tracking a walking person.
Resilience to multipath: the approach should work in moderate multipath environments, meaning outdoor deployments with buildings, cars, trees and people close by.
Low-power, low-cost: the range measurements can be collected using standard sensor network devices with resource constraints such as 8 MHz microcontroller, 9 kHz ADC sampling, 4 kB of RAM memory, 56 kbps radio and no additional specialized hardware.

In this paper, we show how to improve the time effectiveness of radio-interferometric ranging to allow for real-time refresh rates with sufficient ranging accuracy. First, frequent calibration of radio hardware is required to achieve the desired beat frequency, because manufacturing differences coupled with environmental effects may cause up to 50 ppm errors in the emitted frequency. We show that it is possible to completely eliminate time consuming recalibration by measuring the frequency of the interference signal and adjusting the radio settings on-the-fly. As a result, the improved algorithm carries out the calibration in parallel with ranging. Second, we observe that certain types of measurement errors cannot be eliminated at the ranging level and propose a novel technique that outputs a set of possible ranges, the true range being one of them with high probability. In turn, our tracking algorithm uses the fact that the true ranges from different nodes correlate and it is able to reject the erroneous measurements. We implement this algorithm on a centralized node (a PC laptop) where all ranging measurements from the individual sensors are routed. We show that this process is superior to the previously described localization algorithm, improving the time required to compute the location by an order of magnitude.

2 Radio-Interferometric Ranging

Our tracking algorithm builds on the radio-interferometric ranging technique introduced in [1] which was shown to provide accurate ranges over large distances using low-cost COTS hardware. The unique feature of this work is that two transmitters A and B transmit unmodulated high frequency sine waves at slightly different frequencies concurrently. The resulting composite interference signal has a low beat frequency and can be analyzed on resource-limited hardware using the received signal strength indicator (RSSI) signal. The phase of the composite signal, when measured at receiver C, depends on the phase difference of the two transmitters and their distances from C. Typical low-cost hardware does not allow for phase control at the transmitters, therefore, one more receiver D is used to analyze the same composite signal. The relative phase offset of C and D depends only on the distances between the four nodes and can be expressed as follows (see [1] for the formal proof):

$$d_{ABCD} \bmod \lambda \ = \ \varphi_{CD} \frac{\lambda}{2\pi}, \tag{1}$$

where λ is the wavelength corresponding to the carrier frequency, φ_{CD} is the relative phase offset of C, D, and d_{ABCD} is the interferometric range (q-range) given by the equation $d_{ABCD} = d_{AD}+d_{BC}-d_{AC}-d_{BD}$, d_{XY} being the Euclidean distance between X and Y.

Therefore, radio-interferometric ranging measures q-ranges, rather than pairwise ranges. Furthermore, the wavelength λ is typically much smaller than the actual distances between the nodes, so Eq. (1) can have many solutions. This ambiguity of the q-range solution can be resolved by measuring phase differences φ_{CD_i} at multiple frequencies with wavelengths λ_i, resulting in a system of Diophantine equations with unknowns d_{ABCD} and $n_i \in \mathbb{N}$:

$$d_{ABCD} \ = \ \varphi_{CD_i} \frac{\lambda}{2\pi} \ + \ n_i \lambda_i \quad (i = 1, \ldots, m) \ . \tag{2}$$

The q-ranging algorithm proposed in [1] first instruments two nodes to transmit at multiple radio channels, collects the phase readings from the receivers and then solves (2) for each pair of receivers. The prototype implementation achieved an average localization error of 4 cm for 16 stationary Mica2 nodes covering a 120 × 120 m area.

2.1 Radio-Interferometric Ranging and Tracking

In tracking, we can assume that we only measure q-ranges d_{ABCD} for which we know locations of three out of the four nodes. The q-range equation then reduces to the equation defining a hyperboloid in 3D space with the tracked node being located on the surface of the hyperboloid. If the target node is one of the transmitters, for example A, then both distances d_{BD} and d_{BC} are known and the q-range defines the following hyperboloid with foci C and D:

$$d_{AD} - d_{AC} = d_{BD} - d_{BC} - d_{ABCD} . \tag{3}$$

Similarly, if the target is one of the receivers, for example C, the q-range defines the following hyperboloid with foci A and B:

$$d_{BC} - d_{AC} = d_{BD} - d_{AD} - d_{ABCD} . \tag{4}$$

For a set of n nodes, a single measurement round (i.e. two nodes are transmitting and the remaining $n-2$ nodes are measuring the phase) yields $\binom{n-2}{2}$ pairwise phase differences, and thus $\binom{n-2}{2}$ q-ranges.

It is easy to see that if the tracked node is one of the transmitters, its location can be found at an intersection of $\binom{n-2}{2}$ hyperboloids defined by these q-ranges. Moreover, the receivers (that is, the foci of the hyperboloid) are presumably located at different points in space, so each pair of receivers defines a unique hyperboloid (see Eq. (3)).

In contrast, if the tracked node is a receiver C, only $n-3$ of these q-ranges are relevant for its localization–those defined by the receiver pairs in which C participates. More significantly, each of the hyperboloids corresponding to the $n-3$ q-ranges is uniquely defined by the two fixed foci, transmitters A and B, and the target C. Therefore, all $n-3$ q-ranges define a single hyperboloid. In order to obtain a unique solution for C, multiple measurement rounds with different transmitter pairs are thus required.

We choose the tracked node to be a transmitter because it provides better ranging data per ranging measurement. A drawback of this approach relates to its extension to multiple tracked targets. Such extension would require serialized access to the radio channel by the targets which would decrease the refresh rate proportionally to the number of targets.

3 Approach

The overall approach of inTrack, as shown in Fig. 1, is as follows.

1. The target requests the location calculation on demand. The second transmitter is chosen from the infrastructure nodes and all nodes involved in the measurement time-synchronize using the estimated time on arrival (ETA) primitive [19] to coordinate their transmissions and receptions.
2. For each radio channel, receivers measure frequency and phase of the periodic interference signal by analyzing the signal at the RSSI pin of the radio chip. Each receiver then routes all measured data to a PC-class server. After the routing task is completed, the target requests another localization round.

3. For each pair of receivers, the phase differences are computed for all channels. Consequently, the q-ranging algorithm is executed on the server to calculate q-range for each receiver pair.
4. The location of the tracked node is computed. This algorithm is executed on the server and makes the computed location available remotely on the web.
5. Both Google Earth (see Fig. 7) and our proprietary GUI show the location and the track of the target node on the map in real-time.

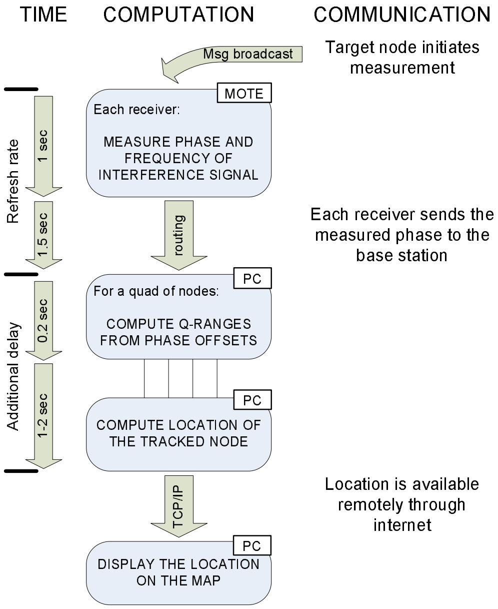

TIME
COMPUTATION
COMMUNICATION
Msg broadcast
Target node initiates measurement
MOTE
Each receiver:
MEASURE PHASE AND FREQUENCY OF INTERFERENCE SIGNAL
Refresh rate
1 sec
1.5 sec
routing
Each receiver sends the measured phase to the base station
PC
For a quad of nodes:
COMPUTE Q-RANGES FROM PHASE OFFSETS
Additional delay
0.2 sec
1-2 sec
PC
COMPUTE LOCATION OF THE TRACKED NODE
TCP/IP
Location is available remotely through internet
PC
DISPLAY THE LOCATION ON THE MAP

Fig. 1. A block diagram of the tracking algorithm illustrating the timing, computation and communication components of the algorithm. The time requirements consist of the refresh rate portion (2.5 sec) which is the time needed to measure and route ranging data from the nodes to the base station, and the additional delay (2 sec) needed for the location computation. Only the phase measurement is currently implemented on the nodes, all other computation is done on a PC.

3.1 Frequency and Phase Analysis

Probably the most time consuming operation in the process of acquiring the frequency and phase of the interference signal is frequent calibration (see [1]). This is required because the frequency at which the transceivers transmit is dependent on the hardware differences and the temperature and voltage variations observed in a real-world deployment of the sensors. The time required for the calibration is unacceptably long and would decrease the refresh rate of tracking. We have enhanced our approach eliminating the re-calibration completely and hence, significantly decreasing the measurement time.

VCO and PLL Self-Calibration To compensate for changes in supply voltage and temperature, the VCO and PLL of the radio transceiver (CC1000 from Texas Instruments, formerly Chipcon) [17]) need to be calibrated. This self-calibration takes 34ms for the transceiver to complete, and has to be repeated every time when a new operating frequency is set.

We found that the calibrated settings for the VCO and PLL do not change significantly for a few hours of continuous operation. Therefore, we self-calibrate the radio chip for all radio channels at start-up, store the calibration values in local memory, and use the recorded values to rapidly set up the radio for consequent radio-interferometric measurements at different frequencies. The price we pay is a) larger initialization time of the node (up to two seconds) and b) a dedicated memory buffer at each node to hold calibration values (up to a hundred bytes to cover the frequency band from 400MHz to 460MHz). This is, however, well worth the achieved 50 % improvement of the channel measurement time.

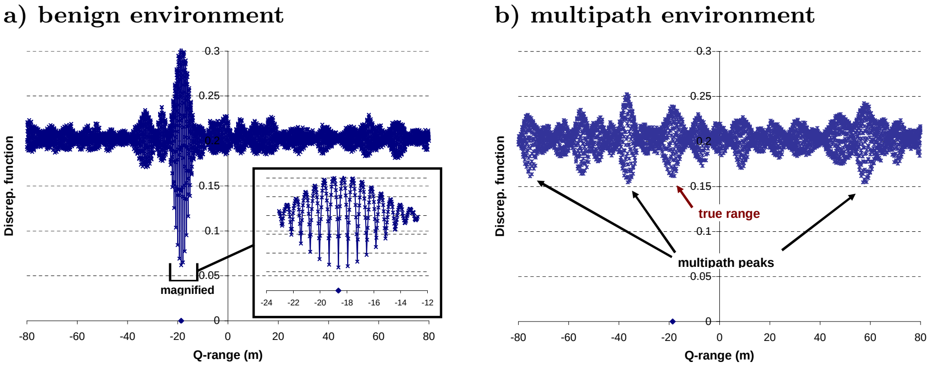
Beat Frequency Calibration (tuning) The algorithm that calculates the phase and the frequency of the interference signal (see [1]) can only analyze signals with frequencies in a relatively narrow range (i.e. 250–450 Hz), due to the resource limitations on the motes. Therefore, it is essential that we can set the frequency difference of the two transmitters with a very fine grained resolution.

However, the errors of up to 50 ppm in the transmit frequency can be expected when using low-cost hardware, which results in an interference frequency out of the analyzable range. Previously, we have proposed a tuning algorithm [1] that is executed before every ranging measurement to compensate for these deviations. As the tuning procedure for a given pair of transmitters takes approximately 3 seconds to finish, this conflicts with the real-time requirements of our tracking service.

Since the output frequency of the radio transceiver is highly sensitive to conditions such as supply voltage and temperature, it is not sufficient to execute the tuning algorithm at application startup only. Instead of time consuming periodic tunings we inserted the following control loop in our algorithm: we monitor the beat frequencies measured by the receivers during q-range measurements and optionally update the tuning values of the transmitters. If the observed frequency is too low (< 300 Hz) or too high (> 400 Hz), the receivers notify the transmitters to update their tuning values accordingly. Consequently, the transmitters are being constantly adjusted and there is no need for the tuning procedure (provided the q-range measurement is initiated frequently enough). The tracking algorithm calculates a new location every few seconds assuring the constant flow of q-range measurements as well as the constant update of the transmit frequencies.

3.2 q-range Computation

The algorithm published in [1] solves the system of Diophantine equations (2) in the following way: it defines a phase offset discrepancy function f for a q-range q as the root mean square error (RMSE) of the measured phase offsets φ_{CD_i}


a) benign environment

b) multipath environment

Fig. 2. a) The global minimum of the discrepancy function f is at the true range. The magnified interval shows that the values of f are close to the global minimum at certain points around the true range (multiples of $\overline{\lambda} = 70$ cm). b) Multipath introduces additional phase measurement errors, resulting in multiple local minima with almost the same value. Global minimum may no longer be at the true range.

from their expected value at q:

$$f(q) = \sqrt{\sum_i (\varphi_{CD_i} - \mathbf{mod}(q, \lambda_i))^2}\ .$$

q-range d_{ABCD} is then defined as the point where f has the global minimum (see Fig. 2a). It has been observed in [2] that the phase measurement errors and the relatively small bandwidth the radio circuitry allows (difference between the maximum and minimum wavelengths is 10 cm) have negative effects on the error distribution of the calculated q-ranges. In particular, the value of the function f is close to the global minimum at small integer multiples of the average wavelength $\overline{\lambda}$ from d_{ABCD} (see magnification in Fig. 2a). Therefore, the distribution of q-ranging errors is not Gaussian, but it exhibits increased error probabilities at small integer multiples of $\overline{\lambda}$ from the mean. The q-ranging error is further influenced by multipath as shown in Fig. 2b. Even though the discrepancy function has a local minimum at the true q-range d_{ABCD}, the global minimum of f can be located far away.

In this paper, we aim to eliminate both the wavelength and multipath related portions of the ranging error (shown in Fig. 2). Since the global minimum of the discrepancy function might be a false solution and hence the true solution might be at a local minimum, we define the output of our ranging algorithm as a set of possible q-ranges rather than a single value. The task of choosing the correct q-range from this set is left to the tracking algorithm. Formally, we define the ***q-set*** S_{ABCD} as:

$$S_{ABCD} = \{q \in \mathrm{I\!R}, \forall i : |q \bmod \lambda_\mathrm{i} - \varphi_{\mathrm{CD_i}}| < \mathbf{windowSize}\}, \tag{5}$$

where A, B are transmitters, C, D are receivers, φ_{CD_i} is the relative phase offset of C, D measured at channel with wavelength λ_i, for $i = 1, \ldots, \mathbf{m}$ and

windowSize specifies how much discrepancy in the phase offset measurements is tolerated.

We would like to point out that if the tracked object is moving during a measurement round, q is not constant in Eq. (5). If the speed of the tracked object is large, or the measurement takes too long (large **m**), the observed variance of q becomes larger than **windowSize**, resulting in $S_{ABCD} = \emptyset$. One could argue that this could be resolved by setting **windowSize** large enough. However, increasing the window size above $\overline{\lambda}/2$ would mean that every range q would satisfy the constraints of Eq. (5). Therefore, we needed to constrain the window size and consequently, the maximum speed of the tracked object as well.

The tradeoff between accuracy and real-time requirements also needs to be considered: using more phase offset measurements to constrain the q-range set (larger **m**) results in smaller q-set S_{ABCD} which requires less computation in the subsequent location calculation and allows for more accurate location estimation. Using less phase measurements, on the other hand, decreases the measurement time resulting in better refresh rate and higher maximum speed.

The choice of parameters **windowSize** and **m** is therefore influenced by the objectives of the tracking algorithm. We wanted to track targets of up to 2 m/s with 2–3 sec update rate with a reasonable accuracy. Our measurements (see Fig. 3) have shown that good accuracy can be achieved using **m** $=$ 11 channels. On the other hand, to keep the size of the calculated q-sets reasonably small, we had to limit **windowSize** to 25 cm. These two parameters are in sharp contrast: since the measurement time is 40 ms for a single channel, the q-range can change by almost 1 m during the 11-channel measurement for a target moving 2 m/s. This is significantly more than the allowed **windowSize** limit and so the calculated q-set will be empty. We solved this problem by calculating q-sets using small subsets of consecutively measured channels, called ***frames***, instead of using the full set of eleven channels. The small number of channels in a frame allows to compute q-ranges for a moving target, and multiple frames together can utilize information measured at all 11 channels, allowing for high accuracy. In practice, we used 4 frames that utilized each of the 11 measured channels. The q-ranges, calculated for different frames, can still differ by up to 1 m because of the moving target. However, the localization algorithm can resolve this by considering larger regions as will be shown in the next section.

3.3 Tracking

Ranging algorithms often have to deal with multiple sources of error that distort the measurements. It is usually possible to study these error sources and calculate more information than just a range from the measured data. This extra information can help to improve the location estimates calculated from a large set of ranging data.

The study of the errors involved in the radio-interferometric ranging enables us to determine q-sets from the measured data. Q-sets are relatively small sets of q-range candidates, the true range being one of them with a high probability. We have shown in Sect. 2.1 that each q-range corresponds to a hyperboloid in

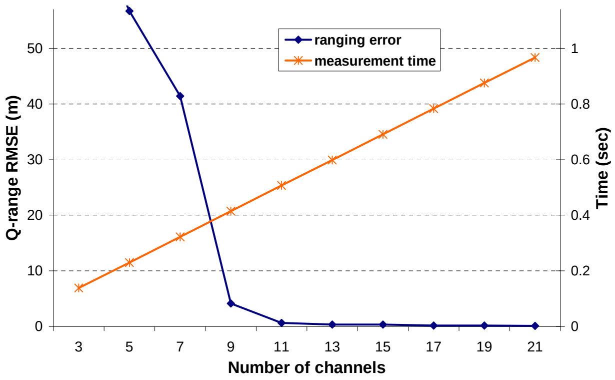


Fig. 3. Radio-interferometric ranging measures the relative phase offsets of two receivers at multiple radio channels. The root mean square error (RMSE) of q-range improves as the number of channels is increased, whereas the time required to measure and transfer the data to the base station grows. The figure shows simulated data with typical measurement noise.

3D and that a single ranging measurement results in multiple q-sets (one for each receiver pair). Our localization algorithm then uses the fact that there is at least one hyperboloid in each of the q-sets that intersects the location of the target. In contrast, it is very improbable that a significant number incorrect hyperboloids intersect at a single point. Consequently, even though the majority of the ranging data is incorrect, the localization algorithm can find the position of the tracked node by finding a region where most hyperboloids intersect.

This is basically a search problem which can be solved algorithmically. Let $S_D = \bigcup S_{ABCD}$ be the set of all q-ranges computed by the ranging algorithm for a target node A. Let $\mathcal{X}$ be a box in 3D space. We define the consistency function $C(\mathcal{X})$ as the number of q-ranges from S_D such that their corresponding hyperboloids intersect box $\mathcal{X}$:

$$C(\mathcal{X}) = \left|\left\{d_{ABCD} \in S_D : d_{ABCD} = d_{XBCD} \text{ for some point } X \in \mathcal{X}\right\}\right|$$

The General Bisection method based on interval arithmetic [18] was shown to find global maxima of nonlinear functions in a computationally efficient way [3]. The algorithm maintains a list of boxes along with the values of the consistency function for each of the boxes. It is initialized with a box that corresponds to the whole search space, the consistency value of which is the number of all q-ranges $|S_D|$. At subsequent steps, the algorithm removes the box with the largest value, splits the box along the longest dimension into two equal boxes, recalculates the consistency function of the two boxes and inserts the new boxes back to the list. As the sizes of the boxes decrease, the consistency value is also decreasing. The algorithm stops, if the consistency function falls under a certain threshold and the center of the smallest box is returned as the location estimate. We allowed for 5% overlap of the boxes to reduce the risk of splitting a large box at the target location, resulting in two smaller boxes containing half of the good data each.

The value of the consistency function for the resulting box is ideally $\binom{n-2}{2}$ times number of frames calculated per receiver pair (which was 4 as discussed in the previous section). In practice, we used 60% of this ideal value as the consistency function threshold. Further, we accepted the computed box as a valid solution, only if all its sides were smaller than 2 m.

3.4 Tradeoffs

Tracking algorithms approximate continuous change of the location of the tracked object with discrete location measurements. Therefore, the design of these algorithms needs to balance the frequency of the location estimates with the amount of collected localization data and the computational power of available hardware. In other words, real-time can be traded off with the location accuracy and the price of the system. This tradeoff is especially important in the resource constrained domain of wireless sensor networks.

The accuracy of the radio-interferometric ranging depends on the number of channel measurements made for a fixed pair of transmitters. Simulation results in Fig. 3 show, that the ranging error can be significantly improved if more channels are measured. However, it requires more time to measure more channels.

More powerful hardware allows for faster sampling, computation, and radio transmission of the data. Therefore, more data can be measured at the same or even shorter time improving both accuracy and refresh rate. However, more powerful hardware is typically more expensive and consumes more power, increasing the cost and decreasing lifetime of the tracking system.

Lower speed of the tracked object means more stable q-range during the measurement resulting in higher accuracy. However, if the speed goes over a certain limit, the tracking algorithm fails to return a location at all (see Sect. 3.2).

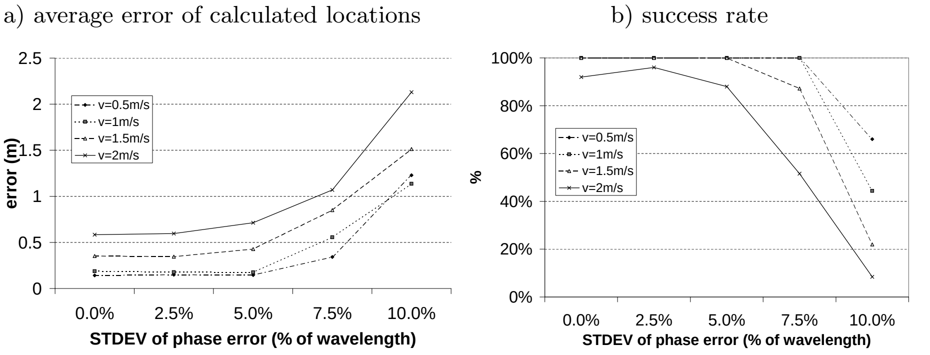
4 Results

4.1 Simulation Results

In this section, we present simulation results to evaluate the tracking algorithm. In particular, we investigate the effects of target speed, as well as the sensitivity of the calculated positions to measurement noise.

Parameters The simulator generates phase and beat frequency measurements emulating the target node moving along a trajectory being tracked by a set of infrastructure nodes. The inputs of the simulator are the infrastructure node locations, target speed, trajectory of the target, noise pattern, transmitter pairs, channel frequencies and channel measurement time.

Assumptions The trajectory of the target is defined by a sequence of waypoints, the simulated trajectory between waypoints is a line. The target speed is assumed to be constant. These assumptions allow for easy implementation and provide a sufficiently detailed model for our purposes.


a) average error of calculated locations

b) success rate

Fig. 4. Target locations are calculated for randomly generated set of 250 target positions and velocity vectors for each pair of speed and phase error deviation. a) shows the average error of the calculated locations and b) shows the percentage of simulated target locations where the tracking algorithm returned a location.

The simulator assumes that a phase measurement at given radio channel is instantaneous, that is, the phase does not change during the measurement. In practice, measuring one channel takes about 40 ms which corresponds to 8 cm position change at a speed of 2 m/s, hence the above assumption is justifiable.

For simplicity, the simulator uses a Gaussian noise model. The noise is added to the generated phase measurement data, is assumed to have zero mean and its variance is a parameter. Our experimental observations show that these are reasonable assumptions.

In practice, message loss and communication delays do influence the results computed by the tracking algorithm. Evaluating the effects of data loss is beyond the scope of this paper, so we assume no data loss in the simulation.

Approach To generate simulated phase measurements, we use Eq. (1), a formula that relates a distance aggregate for a quad of nodes and the difference of phases measured at two receivers. We evaluate Eq. (1), where A is the instantaneous position of the target at the time of the measurement, point B is the position of the second transmitter, point C is the infrastructure node that is measuring the phase offset and point D is a dedicated reference infrastructure node. This way, the simulated phase measurement is a phase offset between node C and the reference node. Since the phase values are never used directly in the tracking algorithm, just pairwise phase differences, having the same additive constant (the phase measured by the reference node D) in all phase measurements is indifferent. Therefore, the simulated values are valid.

To calculate the target position, the tracking algorithm uses phase data measured at different frequencies, and thus at different time instances. Therefore, there are multiple ground truth positions belonging to a calculated target position. In order to relate the output of the tracking algorithm to a single reference

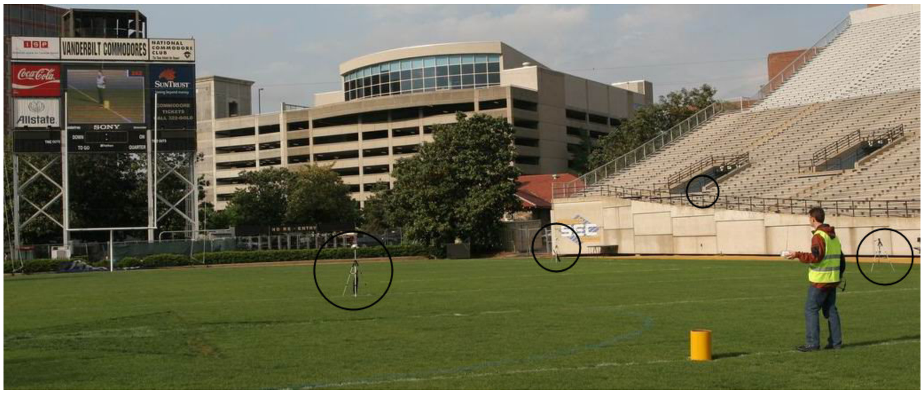


Fig. 5. Our experimental setup at the Vanderbilt football stadium. 12 infrastructure nodes (visible ones are circled on the picture), were positioned on the tripods on the field and on the bleachers in the stands.

position, we consider the centroid of the positions at which the phase measurement data was collected as the ground truth.

The generated phase and frequency measurement are input to the tracking algorithm. The calculated target positions are compared to the ground truth and the average error is calculated.

Results We ran the simulation in a 80×90 m area covered by 12 infrastructure nodes, which is identical to our experimental setup described in the next section. The phase measurement time was set to 40 ms. The phases were measured at 11 channels in the 400–460 MHz frequency range. We used a 4-second position refresh period. For a particular set of simulation parameters, we used the same trajectory defined by randomly generated waypoints within the area of interest. We varied the standard deviation of the Gaussian phase measurement error between 0% and 10% of the wavelength and the target speed between 0 and 2 m/s. A typical standard deviation of the phase error measured by our hardware varies between 3 and 5%, depending on the environment.

The simulation results are shown in Fig. 4. As expected, position errors increase as the speed or the phase measurement errors increase. When the target speed is smaller than 2m/s and there are no significant phase errors present, the position errors are below half a meter. The average position errors exceed one meter only for a large (10%) phase error. The negative effect of large noise combined with large speed is that the tracking algorithm is not able to compute any location for a significant number of measurements, as discussed in Sect. 3.3.

It is interesting to see that for speeds less than 1.5 m/s, the effect of phase measurement error of 5% and under is very similar. The implications of this finding are twofold. First, this means that our tracking algorithm is resilient to errors up to 5% of the wavelength. Second, this result tells us that increasing the precision of the phase measurement, or using a more precise numerical rep-

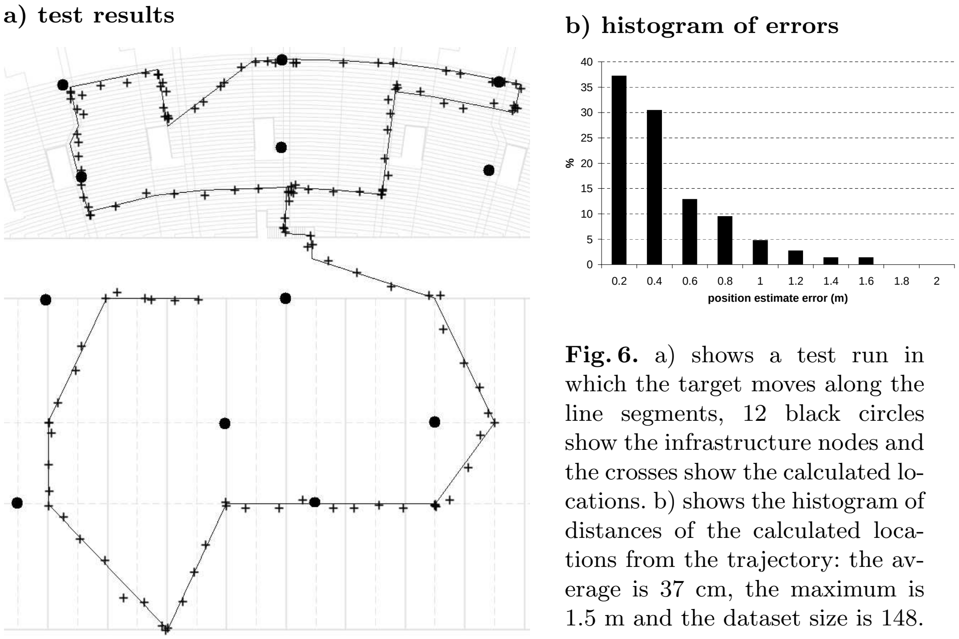


Fig. 6. a) shows a test run in which the target moves along the line segments, 12 black circles show the infrastructure nodes and the crosses show the calculated locations. b) shows the histogram of distances of the calculated locations from the trajectory: the average is 37 cm, the maximum is 1.5 m and the dataset size is 148.

resentation of the phase data such that the errors are bounded by 5% of the wavelength would yield no improvement in the tracking results.

4.2 Experimental Results

We have tested our system at the Vanderbilt football stadium using XSM motes by Crossbow [16] (a version of the Berkeley Mica2 mote in a weatherproof package), running TinyOS [9]. 12 infrastructure nodes were placed at known locations, 6 of them on the field and the other 6 in the stands on one side of the stadium. The trajectory, spanning the whole area, was defined by a series of waypoints. We assume that between the points, the tracked object travelled approximately along a straight line at a constant speed between 1.5 to 2 m/s. The test area that we covered was approximately 80 × 90 m. We achieved a location refresh rate of 3 seconds with an additional 2 second latency. The calculated location and trajectory of the target were available on the internet through Google Earth in real-time. Fig. 5 shows the photo of our experimental setup, Fig. 6 presents the results of one of the test runs in our Java tool and Fig. 7 shows a snapshot of Google Earth.

The elevation map of the area was available to us, so we were only interested in the 2-dimensional location error. For every calculated position, we computed its distance from the closest point on the trajectory and show the corresponding histogram in Fig. 6b.

This evaluation metric, however, underestimates the actual 2-dimensional location error as we do not have any information on *where* on the trajectory the

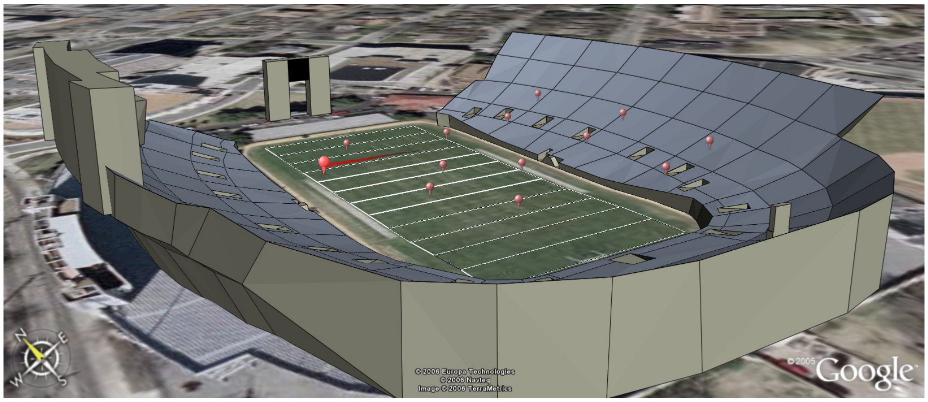


Fig. 7. The current location and trajectory of the tracked node could be accessed through Google Earth in real-time. A large balloon shows the locations of the tracked node while smaller ones indicate the twelve infrastructure nodes.

target was at the moment of the corresponding measurement. By simulation, we found that the distance from the trajectory is typically 65% more than the actual location error: therefore, we estimate that the average location error was 0.61m in our experiment. This result aligns well with the simulated data: if the standard deviation of the phase errors is between 2.5% and 5% of the wavelength, and the target speed is 1.5 m/s to 2 m/s, the location errors range from 0.34 m to 0.71 m.

5 Related Work

Both node localization and tracking are well established areas in sensor networks research. These algorithms typically rely on distance measurements to estimate the range between neighboring nodes. Several techniques can be used to generate these measurements, including time-of-flight, time difference of arrival, angle of arrival, phase and/or signal strength measurements, and others. A comprehensive survey of different approaches can be found in [7].

The Global Positioning System is perhaps the most widely published localization system [10]. GPS provides reliable worldwide coverage, but is considered to be a coarse-grained, high-cost and power inefficient solution in sensor networks. State of the art low-power GPS receiver ICs cost around $50, have an accuracy of 3–5m and power consumption in the 50-100mW range.

Among the typical sensor network solutions, time-of-flight ranging that combines acoustic and RF signals tends to give much better results than RF received signal strength (RSSI) because of the high variance of RSSI [8, 14]. Ultrasound and more recently broad ultrasound measurements have become popular, especially for fine-grained indoors localization and tracking [4, 13, 6]. The system presented in [5] avoids using range measurements by using only connectivity between nodes at the price of decreased accuracy.

Few of the published tracking algorithms have been demonstrated at scale. The system published in [11] is capable of 1 Hz location update rate with a few centimeter average error, but the experimental setup uses 6 anchor nodes to track the target within a 1.5×2 m area. The connectivity-based algorithm in [5] covers a somewhat larger area (10×10 m) with 4 anchor nodes and achieves a 2-second update rate and an accuracy of 1.83 m.

Most tracking algorithms assume the existence of fixed nodes with known locations. The errors of these locations can significantly degrade the accuracy of the calculated target location. Pathirana et al. [12] have shown that the known trajectory of the target can be used to discover and improve the location estimates of the fixed nodes. Taylor [15] have improved this to the point where they perform localization and tracking simultaneously, without the knowledge of the mobile node's trajectory. However, their system relies on acoustic ranging and hence, it requires a fairly dense network.

6 Discussion

We have introduced inTrack, a cooperative tracking system that is able to locate a moving sensor node with high accuracy over large areas. The system can tolerate measurement errors typical in moderate multipath environments. Moreover, inTrack has been implemented on COTS mote hardware, making its deployment economically viable.

The experimental evaluation shows $2-3$ second update rate and 0.6 m average error of the location estimates with the target moving up to 2 m/s within a 80×90 m area. We have generated a large number of random positions, speeds and directions in our simulator and verified the performance of inTrack in much larger scale than was experimentally feasible.

One weakness of inTrack is that it can track only a single target in its current implementation. Multiple targets would have to serialize their access to the inTrack system and the update rate of the locations would decrease. One possible solution is to make the target node a receiver rather than a transmitter. While our prototype uses the XSM motes that cost several hundred dollars, we envision that such a tracking system, assuming mass production of specialized hardware, can work as a cheaper and more accurate version of GPS in such geographically constrained areas as sports stadiums or amusement parks.

Currently a significant portion of our code runs on a PC, namely the computationally expensive search that finds intersections of hyperboloids. A solution could be to find these intersections analytically, enabling the implementation of the algorithm on resource constrained hardware. Finally, we plan to improve the current speed limitation of the inTrack system.

References

1. M. Maróti, B. Kusý, G. Balogh, P. Völgyesi, A. Nádas, K. Molnár, S. Dóra, and A. Lédeczi: Radio-interferometric geolocation. *in Proc. ACM 3rd Conference on Embedded Networked Sensor Systems (SenSys)*, San Diego, CA, USA (2005).

2. B. Kusý, A. Lédeczi, M. Maróti, and L. Meertens: Node-density independent localization. *In Proc. 5th International Symposium on Information Processing in Sensor Networks (IPSN'06)*, Nashville, TN, USA (2006).
3. A. Lédeczi (et. al): Countersniper System for Urban Warfare. *ACM Transactions on Sensor Networks*, Vol. 1 (2005), 153–177.
4. M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper: Implementing a Sentient Computing System. *Computer*, vol. 34, no. 8, pp. 50–56, Aug. 2001.
5. N. Bulusu, J. Heidemann, and D. Estrin: GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28-34, Oct 2000.
6. M. Hazas and A. Hopper. Broadband Ultrasonic Location Systems for Improved Indoor Positioning. *IEEE Transactions on Mobile Computing*, volume 5, number 5, pages 536-547, May 2006.
7. J. Hightower and G. Borriello: Location Systems for Ubiquitous Computing. *Computer*, vol. 34, no. 8, pp. 57-66, Aug. 2001.
8. J. Hightower, R.Want, and G. Borriello: SpotON: An indoor 3D location sensing technology based on RF signal strength. UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA, feb 2000.
9. J. Hill and R. Szewczyk and A. Woo and S. Hollar and D. Culler and K. Pister: System architecture directions for networked sensors. *in Proc. of ASPLOS 2000*, Cambridge, MA (2000).
10. B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins: Global Positioning System: Theory and Practice, 4th ed. *Springer Verlag*, 1997.
11. D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. *in Proceedings of ACM Sensys'04*, Nov 2004.
12. P. Pathirana, N. Bulusu, S. Jha, and A. Savkin Node localization using mobile robots in delay-tolerant sensor networks. *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, Jul/Aug 2005.
13. N.B. Priyantha, A. Chakraborty, and H. Balakrishnan: Cricket Location-Support System. *Proc. Sixth Intl Conf. Mobile Computing and Networking (MobiCom)*, Aug. 2000.
14. A. Savvides, CC. Han, and M. Srivastava: Dynamic finegrained localization in ad-hoc networks of sensors. *In 7th ACM Int. Conf. on Mobile Computing and Networking (Mobicom)*, pages 166-179, Rome, Italy, July 2001.
15. C. Taylor, A. Rahimi, J. Bachrach, H. Shrobe, and A. Grue: Simultaneous Localization, Calibration and Tracking in an Ad Hoc Sensor Network. *In Proc. 5th International Symposium on Information Processing in Sensor Networks (IPSN'06)*, Nashville, TN, USA (2006).
16. P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. *In Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, Special track on Platform Tools and Design Methods for Network Embedded Sensors (SPOTS), April 2005.
17. Chipcon: CC1000 Product Information. 2004, http://www.chipcon.com.
18. C. Hu, S. Xu, X. Yang: A Review on Interval Computation – Software and Applications. *Int. J. of Computational and Numerical Analysis and Applications,* Vol. 1, No. 2, pp. 149–162, 2002.
19. B. Kusy, P. Dutta, P. Levis, M. Maroti, A. Ledeczi, and D. Culler: Elapsed Time on Arrival: A simple and versatile primitive for canonical time synchronization services. *International Journal of Ad Hoc and Ubiquitous Computing,* January 1, 2006.