
Ultrasonic Sensor Network: Target Localization with Passive Self-Localization

Ashutosh Saxena*
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
asaxena@stanford.edu

CS229
Project Report
Prof. Andrew Ng
<http://www.stanford.edu/class/cs229>

Abstract

A flying robot is localized using ultrasonic sensor network distributed on the ground. This requires localization of sensor nodes first. In absence of RF link between target and sensors, weighted Multi-Dimensional Scaling (MDS) with Gradient Descent is used to localize the sensors from outlier free data obtained from generating a locally linear manifold. After initialization, an Extended Kalman filter (EKF) is used to estimate robot's position alongwith other unknown parameters. Outliers, which are numerous because of large distances and noise involved, are detected by checking innovation to be within range. Variable length observation vector helps select the good sensor observations from the available many.

1 Introduction

Absolute positioning of flying robots is extremely important for free range path tracking, reactive navigation, as well as automatic control [1]. With flying robots, localization systems using inertial sensors and GPS suffer from problems in some scenarios like inverted flights and aerobatic maneuvers [1]. Using computer vision suffers from slow update rate.

Deploying ultrasonic sensor networks to solve this problem offers an attractive solution, providing high update rate, high accuracy and robustness using redundancy in the sensors. Each sensor is equipped with an ultrasonic receiver, some computing power, and a wireless link to central computer. A pulse emitted from the transmitter on the flying robot, is received by some of the sensors who report the pulse arrival time to the central computer. In absence of RF-link between ultrasonic transmitter and sensors, time-of-flight (TOF) information is not available, and the position is estimated using pulse arrival times.

Ultrasonic sensors have been use in localization of robots [2,4,5,6]. Problems addressed include estimating variable speed of sound, and combining data from different types of sensors,etc. However, in almost all of them, the following problems are

*Ashutosh Saxena (*ashutosh.saxena@ieee.org*). Project being pursued under Prof. Andrew Ng (*ang@cs.stanford.edu*) in AI-Robotics Lab.

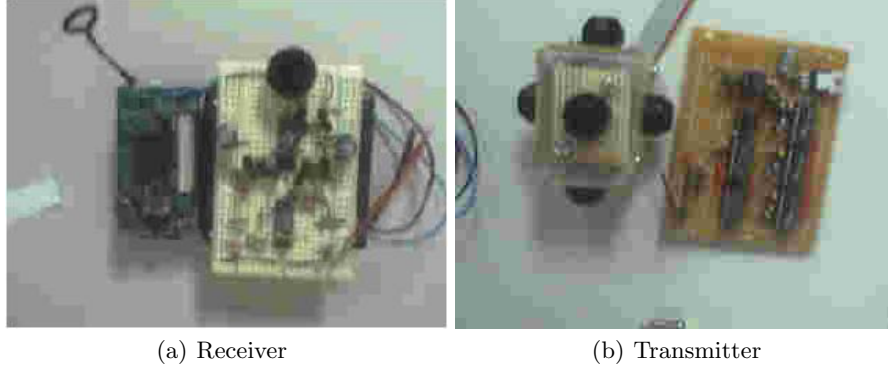


Figure 1: Receiver and Transmitter circuits

not addressed: a) Very unreliable arrival times because of large distances involved and noise, resulting in a large number of outliers, b) Unknown ultrasonic sensor node locations, c) No RF-synchronization between transmitter and receiver. Using redundancy from higher number of sensors, allows to reliably estimate the position if the robot in these scenarios.

Most sensor network algorithms require knowledge of sensor node locations, which is not always practical. Network localization algorithms exist, but some of them need special beacon nodes with known locations, and others need nodes to communicate with each other using sound and RF signals. Passive sensor network localization techniques are promising. Bayesian network [6] approach to passive localization is not applicable in practical scenarios of single target localization because the pulses are not well dispersed (in fact they lie on a particular trajectory) and only a small subset of nodes receive the pulses.

We use Weighted Multi-Dimensional Scaling (WMDS), with minimization over pulse arrival time as well. Outliers are removed by finding a 3-dimensional locally linear [3] manifold in a subset of space of arrival times from all the sensors. Once the sensors are localized, the algorithm switches to Extended Kalman Filtering (EKF) mode for position estimates of the target. A variable length observation vector is used and consistent innovations from each subset of available observed arrival times are used to update the position.

2 Localizer Structure

The localization system consists of M beacons (6 in our specific experiment), and a transmitter array placed on the flying robot which sends a 40kHz ultrasonic pulses (Figure 1(b)) at every 20 ms. Note that there is no RF-Synchronization between the flying robot and the sensor nodes. The sensor nodes are located at \vec{s}_i with $i = 1, \dots, M$. The nodes are placed on uneven ground with approximately known locations \vec{s}_i^0 .

The sensor nodes ((Figure 1(a))) consist of an ultrasonic receiver, signal from which is amplified, filtered and rectified using low-noise analog circuitry and fed to sensor microprocessors which have adaptive pulse energy and noise energy thresholds to detect pulses. The time of arrival of pulses alongwith pulse energy and width is reported to the central computer over wireless network (using Crossbow radio).

The large distances involved (50 ft.) and the noise from flying robot (helicopter) result in poor reception of pulses at the ultrasonic sensors. There are numerous missing pulses and erroneous pulses. The amplitude of the pulse received varies a lot because of many factors like transmitter and receiver angle, shadowing, multi-path interference, etc. Although it has some correlation with the distance from transmitter, the amplitude of the pulse cannot be taken as a feature for distance. The arrival time of the pulses is accurate (to $150\mu\text{sec}$), and is taken as the feature vector for observation.

3 System Model

For the n^{th} pulse, the arrival time at sensor m ($= 1, \dots, M$) is given by

$$t_m(n) = t_0 + nT + D_m(n)/c \tag{1}$$

$$D_m(n) = \|x(n) - s_m\| \tag{2}$$

$$D_{k_1}(n) - D_{k_2}(n) = c(t_{k_1}(n) - t_{k_2}(n)) \tag{3}$$

where, T is the beacon firing cycle time (variable), t_0 is the firing time of first pulse, c is the speed of sound in air (variable but can be modeled). $D_m(n)$ is the euclidean distance of the target when n^{th} was fired from sensor m . Note, that in third equation, taking differences of the distances between two sensors results in cancellation of firing time of the pulse. Thus, if we are using more sensors, then the time-of-flight is not needed.

4 Algorithm

The localization process consists of two parts

1. : Pulse Detection: The noise level and the pulse energy level vary. To take care of this, and ensure that low energy pulses are also detected, we put a adaptive threshold based on total signal energy and pulse energy.

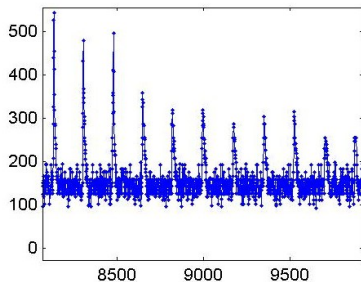


Figure 2: Received signal from 10 ft at the Sensor

2. Sensor Localization: Given approximate location of the sensors placed randomly on the field and unknown position of the target, the task is to accurately find the position of the sensors. There are a lot of outliers in the data because of false pulses and missing pulses.
3. Realtime Target Localization: Once sensor node locations have been determined, the position of the target should be determined in realtime. Only

some of the sensors maybe giving reliable observations, and it is may not be known apriori if the observations are reliable or not.

5 Sensor Node localization

The system equations can be looked at as a 3-dimensional manifold in M-dimensional observation space. The intrinsic free variables are $x = (x, y, z)$ positions of the target. Since, the equations are well-behaved when we linearize them, the trajectory is locally linear. Further, x changes as a result of a target (helicopter) moving, hence $x(n)$'s are closely spaced. Under these conditions, we fit a Locally Linear Manifold [3]. Locally Linear Manifold finds nearest neighbor from a candidate set of neighbors based on how well neighbors construct the point. After repeating this for each point, a outlier free manifold is formed. This presents a novel application of this algorithm in removing outliers from sensor data. This automatically chooses the points for which all the sensor readings are valid, and removing outliers. This can also be extended to the case when only a few sensor readings are valid by fitting manifolds in subspace spanned by rest of the sensors. In this paper, we restrict ourselves to the scenario when all sensor readings are valid (Figure 5).

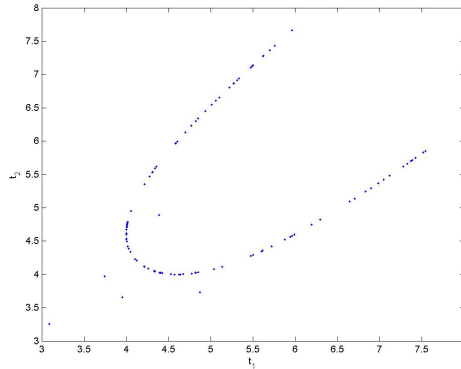


Figure 3: 1-D manifold in 2-D space for 2 sensors and robot moving along a straight line

Another way to look at it is N equations in the following unknown fixed variables: $t_0(0)$, s_{ix}, s_{iy}, s_{iz} (total $3M + 1$ parameters). There are $P \leq MN$ equations, and $3(M - 1) + 3N$ unknowns, and equations can be solved [6], if $P \geq 3M + 3N - 3$. Now, once we remove the outliers, we have pairwise distances (shifted by a additional constant t_0) between $S \leq N$ valid positions of the target and M sensors (all the target positions may not have resulted in valid readings at all the sensors). However, we do not have distances between one sensor to another, and between one target location to another.

We have

$$D(x(k), s_i) = c(t_i(k) - t_0(0) - kT) \quad (4)$$

View $t_0(0)$ as a paramter (and with known T). If we correctly estimate the parameter, we will get pairwise distances between each sensor and robot location in R^3 , by subtracting t_0 from the scaled arrival atime observations t_i . If all the sensor positions and the valid target locations are considered as a vector of positions in R^3 , then

$$z = \begin{pmatrix} s \\ x \end{pmatrix}$$

And we have a matrix of distances $D = zz^T$ in which off-diagonal block matrices are $D_{xs} = xs^T$, and the diagonal matrices $D_{ss} = ss^T$ and the $D_{xx} = xx^T$ are not known

$$\begin{pmatrix} D_{ss} & D_{sx} \\ D_{xs} & D_{xx} \end{pmatrix} + t_0$$

Now, we can apply weighted Multi-Dimensional Scaling (MDS) with $t_0(0)$ as the parameter, and minimize the error by varying $t_0(0)$ according to an iterative method (Gradient Descent). A version of weighted MDS (iterative) is used, with weights corresponding to D_{xs} set to zero.

6 Target Localization

We use Extended Kalman Filter (EKF) for estimating the n -dimensional state of the system, denoted x , given a set of measurements in m dimensional vector y .

6.1 State vector for EKF

The state vector consists of robot position (x, y, z) , and the speed of sound (c) .

$$x = (x \quad y \quad z \quad c)$$

6.1.1 Variable length Observation vector

The observation vector is given by

$$y = (t_2 - t_1 \quad t_3 - t_1 \quad \dots \quad t_M - t_1)$$

If a particular sensor knows that it has not received the pulse, we do not include that particular sensor in that state update, removing that arrival time from the observation vector, and the corresponding matrices.

The number of arrival times required for estimating the position is 4. We have to choose 4 out of 6 observations. This gives us a maximum of $\sum_{i=4}^6 {}^6C_i = 22$ choices for the observation vector to choose from. The following heuristic will be used to optimally choose the observation vector

1. Choose an observation only if its amplitude is above threshold θ_1 . If the number of observations chosen L_N are less than 4, then choose the top 4 if they are above threshold θ_2 , where $\theta_2 < \theta_1$.
2. Reduce the number of observations, if the innovation exceeds a threshold θ_3 .
3. Compute the state covariance matrix with L_N , and with the highest $L_N - 1$ observations. If the error goes down, compute with the highest $L_N - 2$ observations, and continue till $L_N - i \geq 4$.

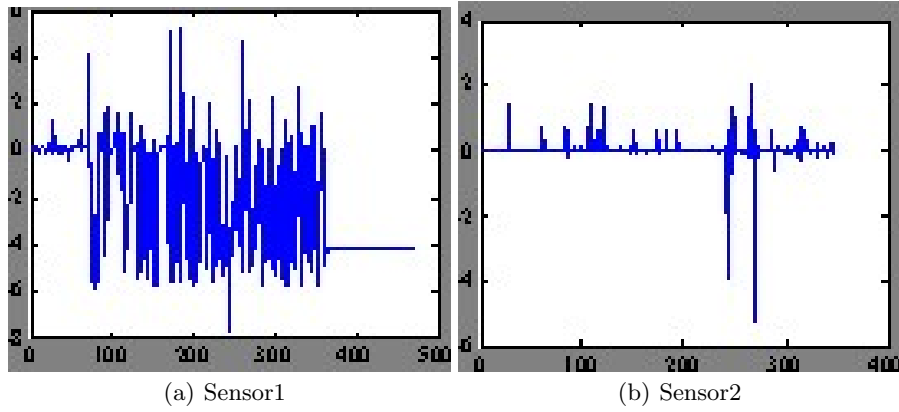


Figure 4: Variable Length Observation vector. First arrival time is spurious after initial few seconds. Second one is good almost all of the times.

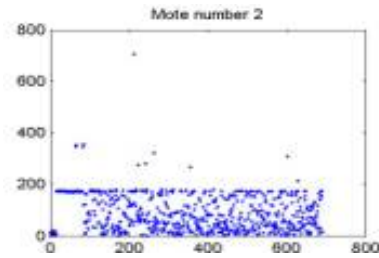


Figure 5: Numerous false peaks are there. Initially, there are no false peaks; but as Helicopter moves far away, the false peaks are detected

6.2 State Transition equations

The State transition equation is

$$x(k+1) = \phi x(k) + w(k+1) \quad (5)$$

where, $w(k+1)$ is the state noise. The Measurement equation is

$$y(k+1) = h(x(k+1)) + v(k+1) \quad (6)$$

$$h_i(x(k)) = (D_{i+1}(k) - D_1(k))/c \quad (7)$$

$$i = 1, \dots, M-1 \quad (8)$$

where, h is a non-linear function, in which non-linearity is not very severe, therefore can be linearized around operating point. $v(k+1)$ is the measurement noise. The EKF filter works well if it is initialized without good values.

6.3 State and error prediction

The variance matrices for state and measurement noise are Q and R are estimated beforehand statistically, to avoid time delay in online estimation. The predicted error covariance matrix P is initialized to a high value because of uncertainties in prediction of state in the beginning. The state prediction is done, predicted error covariance matrix is calculated. Innovation is calculated, and if it exceeds a certain threshold, the observation is rejected. The threshold may be decreased as time passes. The innovation is added to predicted state with a Kalman gain. Then the error covariance matrix is re-computed.

6.4 Spurious arrival time detection

The algorithm may diverge because of spurious arrival times, which cannot be detected in the beginning. In the beginning the error covariance matrices are high because of uncertainties in the state. As the samples start to come, the EKF converges. However, in presence of spurious samples, this may never happen. We need an algorithm for removal of those spurious signals.

Some samples are collected in the beginning, and locally linear neighbors are clustered together as in [3]. These have a high probability of being correct samples are fed to the Kalman filter with weak threshold for spurious sample check. Once, the EKF converges, the samples are fed directly to the EKF.

7 Results

7.1 Simulation Results

The convergence of the Sensor Localization algorithm was checked in simulation. It converged with a high precision.

7.2 Experimental Results

In target localization, the sensor position is assumed to be known. The starting position is also assumed to be known, although, helicopter position was initialized with a wrong value and the Kalman filter automatically corrected it.

7.2.1 Sensor Localization

With 2 sensors, the Locally Linear Manifold algorithm was able to remove outliers.

7.2.2 Estimate of 1-D motion

The error found was 3.5 cm using 2 ultrasonic sensors, and robot moving in 1 dimension.

7.2.3 Estimate of 2-D motion

The arrival times were detected, and the spurious arrival times are removed (Figure 7.2.3).

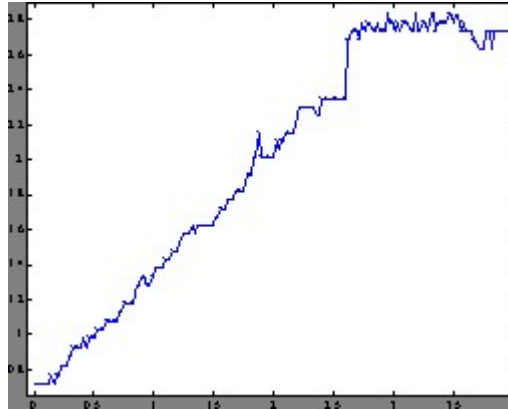


Figure 6: X position of robot moving in a straight line with 3 sensors

7.2.4 Helicopter Localization

The results of localization of helicopter are shown in the figure. The position is shown in meters (Figure 7.2.4). Estimated error is about 15 cm.

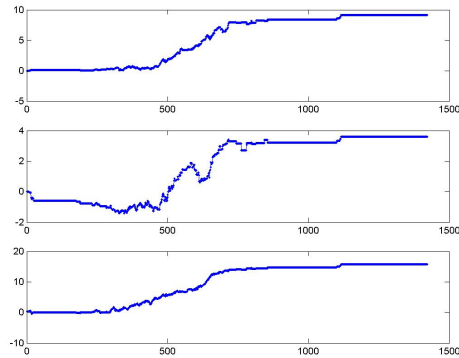


Figure 7: X, Y, Z position of the helicopter in meters with 6 sensors. Output is at 55 Hz

8 Conclusion

The paper will propose a novel location system for robots in 3-dimensions, using ultrasonic sensor network with unknown node locations. This makes the system deployable in scenarios where sensor node localization is not possible, for example, when sensor nodes are dropped by the robot itself. Currently, offline position estimates of 15 cm are available at 25 ft distances. Present system indicates that accuracies of less than 10 cm may be achieved in a 400 sq-ft area for distances upto 50 cm. The algorithm and the system has to be made robust, and operable in real-time.

We also demonstrated a new application of locally linear manifolds in removing outliers, which helps weighted to operate to find sensor node locations.

References

- [1] Andrew Y. Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger and Eric Liang, Inverted autonomous helicopter flight via reinforcement learning, In International Symposium on Experimental Robotics, 2004.
- [2] Michael R McCarthy, et. al., RF Free Ultrasonic Positioning, Proc. IEEE ISWC 2003.
- [3] Ashutosh Saxena, Abhinav Gupta and Amitabha Mukerjee, Non-Linear Dimensionality Reduction by Locally Linear Isomaps, In Lecture Notes in Computer Science (Proceedings of 11th International Conference on Neural Information Processing- ICONIP 2004)
- [4] Ching-Chih, A Localization System of a Mobile Robot by Fusing Dead-Reckoning and Ultrasonic Measurements, IEEE Trans Instr Measurement, vol 47, no 5, Oct 1998
- [5] Lindsay Kleeman, Optimal Estimation of Position and Heading for Mobile Robots Using Ultrasonic Beacons and Dead-reckoning, IEEE 1992
- [6] Rahul Biswas and Sebastian Thrun, A Passive Approach to Sensor Network Localization, unpublished, 2005.