Low-complexity 3D target tracking in Wireless Aerial Sensor Networks

Conference Paper · June 2014

DOI: 10.1109/ICC.2014.6883347

3 authors:

Suleyman Uludag
University of Michigan-Flint

Murat Karakus
Ankara University

Evrim Guler
Bartin University

Low-complexity 3D Target Tracking in Wireless Aerial Sensor Networks

Suleyman Uludag, Murat Karakus and Evrim Guler
Department of Computer Science, Engineering and Physics
University of Michigan - Flint, Flint, Michigan 48502
Email: uludag@umich.edu, {mkarakus, eguler}@umflint.edu

***Abstract*—With the proliferation of the application areas of the Wireless Sensor Networks (WSNs) and the technological advances in sensor electronics, target tracking has become one of the most important functions in WSNs. The overwhelming majority of moving target tracking studies in WSNs has assumed either one-dimensional space (*lineland*), or two-dimensional domain (*flatland*). However, many existing, such as environmental monitoring of rugged terrain, structural health monitoring of buildings, underground and underwater WSN applications, and emerging applications, such as search-and-rescue via Micro Aerial Vehicles, reconnaissance by insect-sized flying robots, need target tracking in three-dimensional space. In this study, we assume a Wireless Aerial Sensor Network (WASN) with conventional low-power and low-complexity features and devise a low spatial and temporal complexity moving target tracking mechanism in 3D based on g-h filter that considers measurement errors. Our prefatory simulations show promising preliminary results for our approach in terms of very low inaccuracy rates between the true location of the target and the forecasts at the sink node as well as negligible estimate errors in terms of Euclidean distance.**

I. Introduction

Wireless Sensor Networks (WSNs) measure ambient conditions in their surrounding environments. The measurements are then transformed into signals that can be processed to reveal some characteristics about phenomena located in the area around these sensors. Although the span, scope, and variety of WSN applications [1], [2] are immense, one frequently occurring common theme for many of the WSN employments is target (moving object) tracking in an area of interest. Figure 1 depicts a high-level representation of a target (the hexagon) moving along a trajectory where detecting sensors report their findings to a sink for further processing.

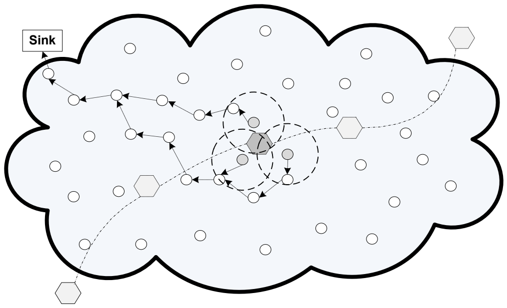

Fig. 1: Tracking a target through its trajectory in WSN and sending the detection to the sink.

A majority of the target tracking approaches reported in the literature are either for one-dimensional space (*lineland*), or for two-dimensional domain (*flatland*). However, there are many current, as well as envisioned future, applications of WSNs that would benefit from target tracking in three-dimensional space. Examples include detecting targets in rubble after natural disasters, sea and underground defense against intrusion, monitoring movements in multi-story buildings, species monitoring, environmental monitoring in rugged terrain, structural health monitoring of buildings, bridges, etc., agricultural monitoring of farms and fields, and airborne target detection by means of low-cost flying sensors. The latter is likely to become an interesting area of *future* research[1] with insect-sized flying robots [3], also known as Micro Aerial Vehicles (MAVs), or other small-scale, untethered, unmanned air devices, such as ornithopters, quadrotors [4], to be used in such tasks as search-and-rescue, environment monitoring and reconnaissance tasks [5]. Further, there is a very promising research direction in which Hybrid Insect Vehicles or cyborg insects are becoming a possibility to harness insects' innate aerodynamic performance with those of extraneous micro-computing and communications capabilities [6].

In this paper, we are proposing a low spatial and temporal complexity, three-dimensional target tracking algorithm for Wireless Aerial Sensor Networks (WASNs). As with the one- and two-dimensional target tracking algorithms, the following tasks must be carried out in a three-dimensional target tracking approach: (1) self-localization of sensors to attain self locations, (2) target detecting by sensors, (3) target localization by the sink, and (4) forecasting future locations of the target by the sink. Our approach assumes an existing localization scheme and a hardware-based detection mechanism, and we concentrate on the latter two tasks in subsequent sections.

Section II provides the context and the related work for our approach. Formal problem definition and formulation are presented in Section III together with our proposed solution based on g-h filter, and its spatial and temporal complexity discussions. Performance evaluation of our scheme is given in Section IV and Section V presents qualitative evaluations, future work and conclusion.

[1]The cost of the flying sensors is of course not as insignificant as the typical sensors but a huge progress in flying objects with sensors on-board has been made in recent years in terms of capacity with ever-declining costs.

II. Background and Related work

An intrinsic and indispensable part of any tracking implementation in WSNs is localization to calculate the locality information within an area of interest with the least inaccuracy possible. Localization can be broken up into multiple categories, as shown in Fig. 2: distributed [7], [8] vs. centralized [9], [10], infrastructure-based [7], [8] vs. infrastructure-less [11], relative [12] vs. absolute positioning [7], range-free [13] vs. range-based [14], and single-hop [15] vs. multiple hops [7].

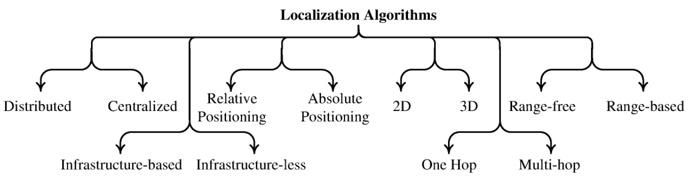

Fig. 2: A general taxonomy of localization techniques.

The localization problem has been extensively studied in the literature, albeit overwhelmingly in two-dimensional domain, such as [19]-[20] and the citations therein. There are far fewer studies on 3D localization, such as [21]-[24].

Tracking [25], [26] makes use of the groundwork established by localization subsystem A taxonomy of tracking approaches for WSNs is presented in Figure 3. A hierarchical tree is formed by the sensor nodes as part of the target tracking approach in tree-based methods [26]-[29]. In terms of the topology, another approach is by means of forming clusters to facilitate collaborative target tracking [30]-[43]. A combination of tree-based and cluster-based has also been proposed in a number of studies [44]-[49]. Another possible categorization of tracking algorithms is based on whether the area is quantized (discrete) [44], [28], [45], [32], [33], [27], [34], [35], [36], [40], [47], [49], [50] or absolute pinpointing is aimed for (continuous) [37], [51], [46], [41]. Finally, some of the targeting approaches assume a constant target velocity [44], [51] while others variable [40], [52].

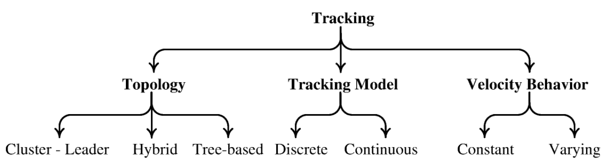

Fig. 3: Classification of tracking techniques in WSNs.

To the best of our knowledge, there appears to be no study for *tracking moving targets in 3D WSNs* even though there are studies in other areas, such as for camera or radar networks [53] with more powerful devices than simple sensors, or in 2D WSNs. The closest one to our problem is in [54] which studied a 3D collaborative target tracking in 3D topological surfaces with the assumption of rather more powerful sensors that are able to infer target locations by means of compute-intensive algorithm without relying on a sink node. Energy consumption of the complexity of the computations was not considered and it appears that the algorithm proposed in [54] may deplete the sensor node batteries quickly.

III. Tracking in 3 Dimensions

A. Preliminaries

Tracking moving targets in three-dimensional WASNs must start with the self-localization of the sensors. The usual process for self-localization is based on the *Trilateration* [55]-[57] method. A sensor will need location information from three nodes which know their locations. Location and distance information from three nodes will enable a node to compute its location by solving a set of linear equations as shown below:

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = r_i^2, \qquad i = 1, 2, 3$$

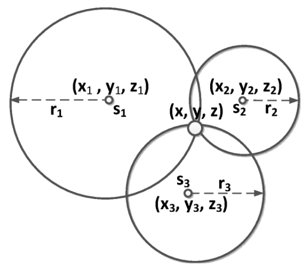

Fig. 4: Self-localization via Trilateration.

where (x, y, z) are the coordinates of the node whose location is to be computed and $(x_i, y_i, z_i), i = 1, 2, 3$ are the three neighbors as depicted in Figure 4. The full details of the Trilateration method can be found in [55]-[57] and is beyond the scope of our paper. We will assume that sensors will have found their locations prior to using our tracking approach as outlined below.

B. Problem Definition and Formulation

Definition 1 (Tracking Target Trajectory in 3D WASNs (3T3D))**.** *Given a set $S = \{s_i | i = 1, 2, ..., n\}$ of sensor nodes that know their own Cartesian Coordinates of $(x_i, y_i, z_i) \in \mathbb{R}^3$, and that can detect and transmit the Spherical Coordinates of a moving target in an area of interest to a designated sink, predict the most likely location for the target at the next time interval at the sink.*

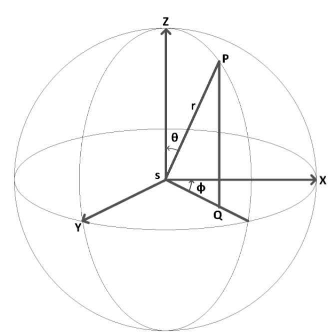

Fig. 5: Measurement parameters by sensor nodes.

Note that the forecast is computed at the sink, not by the individual sensors. In order to provide a solution to the 3T3D problem, we consider a WSN where nodes can detect a moving target and calculate the target's distance r to themselves, the target's azimuth θ (angular distance between North and the object) and elevation ϕ (angular distance between the object and the local horizon)[2], as shown in

[2]Distance and angular measurements by sensors are very well known. For elevation or altitude sensing, inertial or tilt sensors [58], or sensors with inclinometers can be used, especially with advances in Micro-Electro-Mechanical Systems (MEMS) technology, the price and size have been decreasing. See [59] for a sample of various sensor enhancing parts.

Figure 5. We assume that target detecting sensors will transmit these parameters (raw data of (r, θ, ϕ) tuples) to the designated sink, as shown in Figure 1.

Our basic concept regarding how to estimate the location of the target hinge on the idea that target lies on the surface of a sphere having a radius r and centered at (x_i, y_i, z_i) where (x, y, z) contains the three components of the 3D space and i is the node number. To estimate the x, y, z components of the target location, we use the Spherical Coordinate system to calculate the Cartesian Coordinate system components.

We distinguish three cases at the sink to infer the location of the target based on the (r, θ, ϕ) tuples received from sensor(s) to partially offset and mitigate potential measurement errors of the sensor data to the extent possible:

Case 1: A single tuple is received by the sink, that is, only a single sensor detects the target. The sink assumes the tuple as it is to transform the tuple in Spherical Coordinate to Cartesian Coordinate. Specifically, the following formulas are used to compute the Cartesian Coordinates of the target based on the Spherical Coordinates: $x = r \cdot \sin\theta \cdot \cos\phi, y = r \cdot \sin\phi \cdot \sin\theta, z = r \cdot \cos\theta$, where $0 \le r$, $0 \le \phi \le \pi$, $0 \le \theta < 2\pi$, and $r = \|r\|$.

Case 2: The sink receives two tuples from two sensor nodes for the target location. Spherical to Cartesian conversion of Case 1 is applied separately to the two tuples as described earlier. If the tuples do not agree, then the sink will have to interpolate the tuples to decide on the location of the target. Geometrically, given the distance from the two sensors, the exact location of the target lies on the surface of the circle formed by the intersection of the spheres represented by the tuples as depicted in Figure 6. However, exact point on the circle cannot be pinpointed due mostly to the measurement errors. Thus, a simple arithmetic average of $x, y,$ and z coordinates are computed to approximate the target's location.

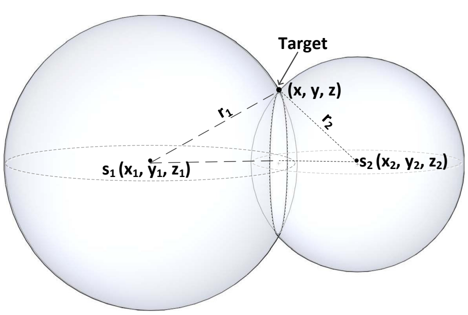

Fig. 6: Intersection of two spheres.

Case 3: The sink receives three or more tuples from sensors detecting the target. In this case, the sink takes the closest three tuples, that is, $n_1 = \min\limits_{\forall i} r_i$, $n_2 = \min\limits_{\forall i \setminus n_1} r_i$, and $n_3 = \min\limits_{\forall i \setminus n_1, n_2} r_i$. Then, we are left with three unknowns and a set of linear equations which can be trivially solved by the sink using matrix algebra as briefly explained below: General equation for a sphere centered at (x_i, y_i, z_i) and having radius of r_i is $(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = r_i^2$. This formula is another form of a second order equation with three unknowns, which is $x^2 + y^2 + z^2 - 2xx_i - 2yy_i - 2zz_i + x_i^2 + y_i^2 + z_i^2 = r_i^2$ Hence, we will have three second order equations:

$$\begin{aligned} x^2 + y^2 + z^2 - 2xx_1 - 2yy_1 - 2zz_1 + x_1^2 + y_1^2 + z_1^2 = r_1^2 \\ x^2 + y^2 + z^2 - 2xx_2 - 2yy_2 - 2zz_2 + x_2^2 + y_2^2 + z_2^2 = r_2^2 \\ x^2 + y^2 + z^2 - 2xx_3 - 2yy_3 - 2zz_3 + x_3^2 + y_3^2 + z_3^2 = r_3^2 \end{aligned}$$

We can degrade those equations to first order linear equations system since the target is on the surface of the spheres and the point where the target is the intersection of these three spheres. By variable substitution in the equations we can generalize them as following:

$$x^2 + y^2 + z^2 + A_i x + B_i y + C_i z + D_i = 0, \qquad i = 1, 2, 3$$

where $A_i = -2x_i$, $B_i = -2y_i$, $C_i = -2z_i$, $D_i = x_i^2 + y_i^2 + z_i^2 - r_i^2$ and $i = 1, 2, 3$ is the index of a node. Now we can equalize these equations since they equal to zero and then we can get rid of these $x^2 + y^2 + z^2$ parts of them. Our new system will be linear equation systems as following:

$$\begin{aligned} (A_1 - A_2)x + (B_1 - B_2)y + (C_1 - C_2)z = (D_2 - D_1) \\ (A_1 - A_3)x + (B_1 - B_3)y + (C_1 - C_3)z = (D_3 - D_1) \\ (A_2 - A_3)x + (B_2 - B_3)y + (C_2 - C_3)z = (D_3 - D_2) \end{aligned}$$

This linear system can be represented in matrix form

$$AX = B$$

which can be easily solved.

We note that there might be no solution to the aforementioned system because of measurement errors. A solution can be approximated in this case by means of the least-square technique.

C. g-h Filter

Section III-B explained how the sink computes the *current* location of the target based on the tuples sent by the sensors. Now, the next step at the sink is to able to map out a projection for the trajectory of the target and make a forecast for its location at the *next sampling time interval*. Such a target tracking problem for fan-beam surveillance radars (for example for ballistic missile tracking) has been extensively studied, especially in the domain of State Estimation and Control Theory [60]-[62]. One very common method by radar tracking application is Kalman Filter, whose essence may be summarized by the following two prediction update equations, assuming a one-dimensional (lineland) approach:

$$\bar{x}_{t+1} = \hat{x}_t + T_t \bar{v}_{t+1} + g_t(y_t - \hat{x}_t) \qquad (1)$$

$$\bar{v}_{t+1} = \hat{v}_t + \frac{h_t}{T_t}(y_t - \hat{x}_t) \qquad (2)$$

where $\bar{v}_{t+1}$ is the filtered or smoothed velocity estimate for time $t+1$ based on the measurements collected up to time t, $\hat{v}_t$ is the estimated velocity from time $t-1$, y_t is the actual observation at t, $\hat{x}_t$ filtered target location or distance estimate at time $t-1$ for time t, T_t is the sampling interval at time t, and $0 \le g_t, h_t \le 1$ are the tunable filtering parameters. The values of g and h provide a relative weight to the target history and the most recent observations. Kalman filter employs a stochastic modeling to infer and dynamically adjust the values of T, g, and h by using spatially and temporally complex computations to fine-tune the estimates.

Since our nodes are low-complexity, low-power and energy-constrained sensors, the Kalman Filter is too much to reasonably expect them to engage in. Instead, our approach, referred to as *g-h filter* in the literature, is based on the underlying smoothing method used by the Kalman Filter by reducing the overhead at the expense of some inaccuracy as shown in the following equations:

$$\bar{x}_{t+1} = \hat{x}_t + T\,\bar{v}_{t+1} + g\,(y_t - \hat{x}_t) \tag{3}$$

$$\bar{v}_{t+1} = \hat{v}_t + \frac{h}{T}\,(y_t - \hat{x}_t) \tag{4}$$

The intuition behind g-h filter is that after the sink computes the location of the target in 3D Cartesian space, it will compute smoothed or filtered projections for the location and velocity of the target. The formulas in Equations 3 and 4 are identical to Equations 1 and 2 except that we are assuming fixed sampling interval T, and filter parameters of g and h to avoid the heavy overhead of the Kalman Filter. See [62] for the detailed derivation of g-h filter that converges to the Kalman Filter formulas as given in Equations 1 and 2.

D. g-h Filter Complexity

The sink will need to run an update of the g-h filter algorithm each time sensors transmit tuples about a detected target. Note that we are assuming a periodic transmission of such updates by the sensors.

In terms of computational complexity, each update of g-h filter only requires four addition and three multiplication operations per dimension, and times three for the three dimensional space. This boils down to a $O(1)$ computational complexity. So, the processing load is negligible on the sink.

As for the spatial complexity, the sink only maintains two variables per dimension or 6 for 3D: the filtered estimate of location for the next sampling interval based on the observations made so far and the filtered velocity for the next sampling interval out of the history of velocity values, both estimated and actual. Note that neither the actual history of observations nor the estimates need to be maintained as the g-h filter formulas (Equations 1 and 2) take the history into consideration through its smoothed and weighted history of computations. See [63] and [62] for more details about computational and spatial complexity of g-h filter.

IV. Performance Evaluation

We have evaluated our approach in extensive simulations. Since our focus of this paper was on the g-h filter-based target tracking approach, the goal of the simulation was to check the accuracy of the target location estimates at the sink by changing the simulation parameters. We have varied both the number of sensor nodes and the area from 200 to 1000. The transmission range values used in the simulations have been between 20 and 100. The kinematic target motion model of the simulations has been based on a smoothed 3D random walk trajectory with constant speed. We have averaged 50 runs for each parameter combination to achieve and exceed statistical significance. The ranges of values used in the simulations for g and h have been $[0.5 - 1.0]$ and $[0.0 - 0.5]$, respectively. Figure 7 shows a representative set of results of our experiments. Figures 7a, 7b, and 7c show the inaccuracy rates of our approach for the three Cartesian Coordinates as computed at the sink node versus the actual locations plotted against the number of sensors, transmission range and the area size, respectively. Very low inaccuracy rates as observed at the sink for the three coordinates are clearly shown. Figure 7a shows an excellent scalability pattern for our approach: When the number of nodes is decreased (that is, the node density is reduced with less potential sensors to detect the target) near constant inaccuracy rates are clearly visible. Figures 7b and 7c display an intuitive and expected behavior for the inaccuracy rate, that is, as the transmission range increases our approach gets better, and as the area size gets larger the accuracy is lost, all other things being equal.

Figures 7d, 7e, and 7f show the inaccuracy of our approach by comparing the actual sink location (*true*), the target location forecast by the sink at time t (*sink*), the estimate for t as estimated at time $t-1$ (*g-h*) by means of both the Euclidean distance and the percentage. Note that the left Y-axis is the inaccuracy of the Euclidean distance in meters while the right Y-axis denote the ratio of the error with respect the the transmission range for the three subfigures 7d, 7e, and 7f. The scalability of our approach as well as its performance under increasing transmission range and area size depict similar patterns in Figures 7d, 7e, and 7f to those of Figures 7a, 7b, and 7c. While the sink's computations for the current time interval is very close to the true location of the target, g-h filter adds about a unit of inaccuracy in terms of Euclidean distance to the true and sink computations. Given the low complexity of our approach, this level of negligible inaccuracy is quite acceptable as a trade-off.

V. Conclusion and Future Work

Studies in the literature on tracking moving targets WSNs have assumed either a one-dimensional or two-dimensional space. Recent advances in sensor technology coupled with emerging new areas of WSN applications necessitate that target tracking should also be considered in three dimensional space. In this study, we have devised a simple mechanism based on a low spatial and temporal complexity filtering mechanism to track a moving *target in 3D* at the sink node by means of measurements supplied by sensor nodes. Our approach is based on the g-h filter as it is known in the field of estimation theory. We have shown by means of extensive simulations that the inaccuracy of our approach under varying parameters is quite low both in terms of percentage in three Cartesian coordinates as well as in Euclidean distance between sink's estimate and the actual location of the target. It is noteworthy that our simulations also show that decreasing the number of nodes do not make much impact on the accuracy of approach which shows some level of indirect fault tolerance against node failures. We believe our approach has shown promising

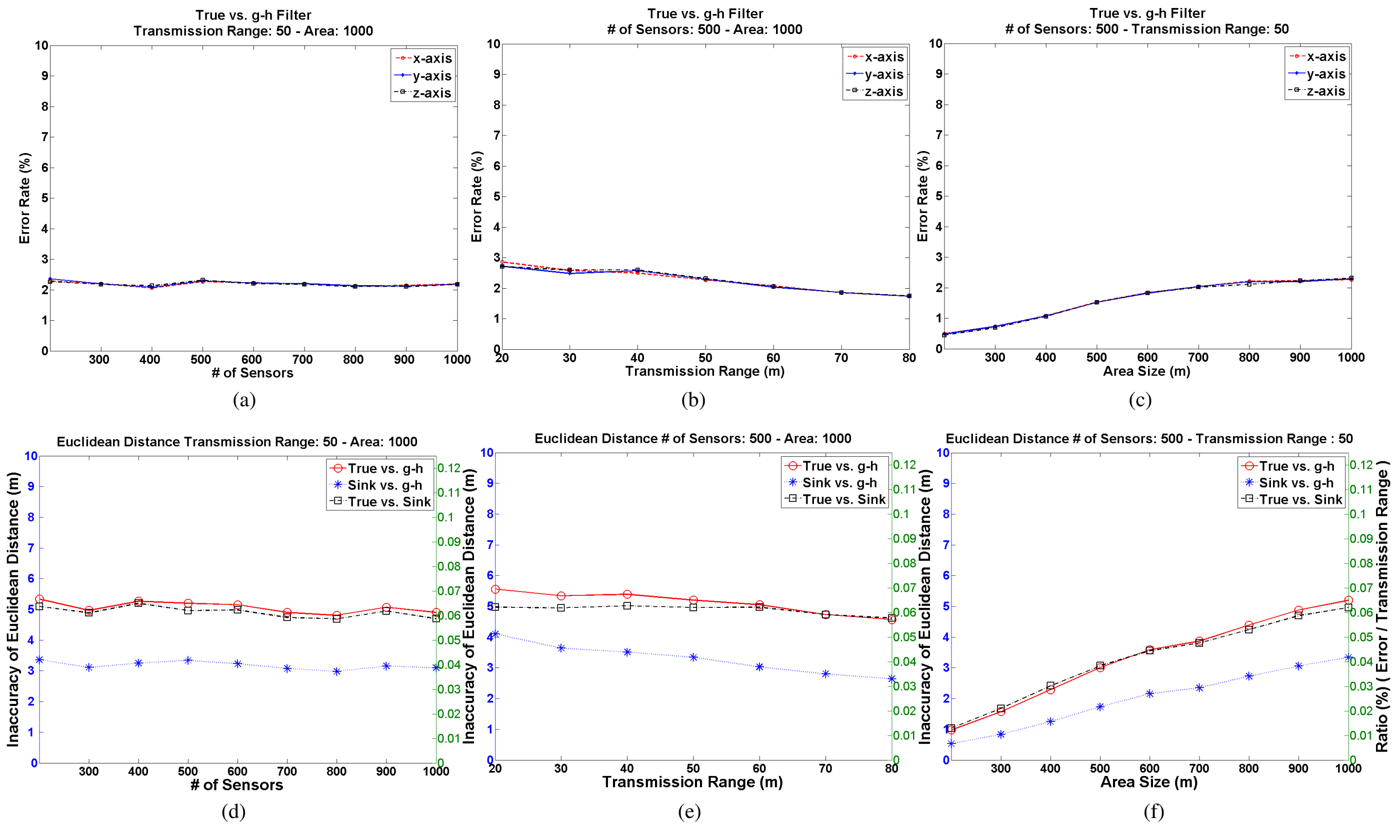

Fig. 7: Accuracy of our approach in terms of percentage difference and Euclidean distance.

results in this initial study but further testing and performance evaluation are needed to assess different dimensions of our target tracking approach as outlined below.

To some certain extent, g-h filter is a simplified version of the well-known Kalman filter which uses dynamic time interval and filter parameters for each measurement interval. It would be interesting to see if a balancing point between g-h filter and Kalman filter can be found to reduce the inaccuracy a little more without incurring too much computational overhead. Further, another approach might be to dynamically change some combination of the filtering parameters on the fly with some feedback coming from the sensors for gauging the accuracy of the previous predictions. Of course, computational, spatial and dissemination overhead should all be considered very carefully. Another extension is to see adapt our approach to tracking multiple targets. We are also curious to see how the accuracy will change if the frequency of sensor measurements and transmissions to the sink is reduced or increased. Related to the measurement frequency, we would like to more formally assess the transmission requirements of our approach. Finally, we would like to study the performance of our approach based on a variety of different target trajectories as we have assumed a rather smooth trajectory in our current study. We have also not considered irregular radios and localization errors. As part of our future work, we will consider noise and measurement distributions for localization and radio errors to assess the impact of false positives/negatives in the overall performance.

Acknowledgment

Part of the funding for this work is generously provided by the Office of Research at the University of Michigan-Flint through an RCAC grant.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Comput. Netw.*, vol. 38, pp. 393–422, March 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, pp. 2292–2330, August 2008.

[3] F. van Breugel, W. Regan, and H. Lipson, "From insects to machines," *Robotics Automation, IEEE*, vol. 15, no. 4, pp. 68 –74, dec. 2008.

[4] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

[5] R. Wood, "Fly, robot, fly," *Spectrum, IEEE*, vol. 45, no. 3, pp. 25 –29, march 2008.

[6] E. E. Aktakka, H. Kim, and K. Najafi, "Energy scavenging from insect flight," *Journal of Micromechanics and Microengineering*, vol. 21, no. 9, p. 095016, 2011.

[7] J. Albowicz, A. Chen, and L. Zhang, "Recursive position estimation in sensor networks," in *Network Protocols, 2001*, Nov. 2001, pp. 35 – 41.

[8] D. Niculescu and B. Nath, "Ad hoc positioning system (aps)," in *GLOBECOM '01. IEEE*, vol. 5, 2001, pp. 2926 –2931 vol.5.

[9] Doherty, L. and pister, K.S.J. and El Ghaoui, L., "Convex position estimation in wireless sensor networks," in *INFOCOM*, vol. 3, 2001, pp. 1655 –1663.

[10] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," ser. MobiHoc '03. ACM, 2003, pp. 201–212.

[11] N. B. Priyantha, A. K. Miu, H. Balakrishnan, and S. Teller, "The cricket compass for context-aware mobile applications," ser. MobiCom '01. ACM, 2001, pp. 1–14.

[12] N. Bulusu, J. Heidemann, and D. Estrin, “Gps-less low-cost outdoor localization for very small devices,” *Personal Communications, IEEE*, vol. 7, no. 5, pp. 28 –34, Oct 2000.

[13] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” ser. MobiCom ’03. ACM, 2003, pp. 81–95.

[14] D. Niculescu and B. Nath, “Dv based positioning in ad hoc networks,” *Telecommunication Systems*, vol. 22, pp. 267–280, 2003.

[15] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The cricket location-support system,” ser. MobiCom ’00. ACM, 2000, pp. 32–43.

[16] E. L. Souza, E. F. Nakamura, and H. A. de Oliveira, “On the performance of target tracking algorithms using actual localization systems for wireless sensor networks,” ser. MSWiM ’09. ACM, 2009, pp. 418–423.

[17] A. Savvides, C.-C. Han, and M. B. Strivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors,” ser. MobiCom ’01. ACM, 2001, pp. 166–179.

[18] T. A. Alhmiedat and S.-H. Yang, “A Survey: Localization and Tracking Mobile Targets through Wireless Sensors Network,” in *PGNet*, 2007.

[19] A. Boukerche, H. Oliveira, E. Nakamura, and A. Loureiro, “Localization systems for wireless sensor networks,” *Wireless Communications, IEEE*, vol. 14, no. 6, pp. 6 –12, December 2007.

[20] C.-L. Wang, Y.-S. Chiou, and Y.-S. Dai, “An adaptive location estimator based on alpha-beta filtering for wireless sensor networks,” in *WCNC 2007, IEEE*, March 2007, pp. 3285 –3290.

[21] V. Yadav, M. K. Mishra, A. Sngh, and M. M. Gore, “Localization scheme for three dimensional wireless sensor networks using gps enabled mobile sensor nodes,” 2009.

[22] C.-H. Ou and K.-F. Ssu, “Sensor position determination with flying anchors in three-dimensional wireless sensor networks,” *Mobile Computing, IEEE Transactions on*, vol. 7, no. 9, pp. 1084 –1097, Sept. 2008.

[23] C. Tian, W. Liu, J. Jin, Y. Wang, and Y. Mo, “Localization and synchronization for 3d underwater acoustic sensor networks,” in *Ubiquitous Intelligence and Computing*, ser. Lecture Notes in Computer Science, J. Indulska, J. Ma, L. Yang, T. Ungerer, and J. Cao, Eds. Springer Berlin / Heidelberg, 2007, vol. 4611, pp. 622–631.

[24] L. Zhang, X. Zhou, and Q. Cheng, “Landscape-3d; a robust localization scheme for sensor networks over complex 3d terrains,” in *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, Nov. 2006, pp. 239 –246.

[25] S. Bhatti and J. Xu, “Survey of target tracking protocols using wireless sensor network,” in *ICWMC ’09*, Aug. 2009, pp. 110 –115.

[26] H.-W. Tsai, C.-P. Chu, and T.-S. Chen, “Mobile object tracking in wireless sensor networks,” *Comput. Commun.*, vol. 30, pp. 1811–1825, June 2007.

[27] W. Zhang and G. Cao, “Dctc: dynamic convoy tree-based collaboration for target tracking in sensor networks,” *Wireless Communications, IEEE Transactions on*, vol. 3, no. 5, pp. 1689 – 1701, Sept. 2004.

[28] C.-Y. Lin, W.-C. Peng, and Y.-C. Tseng, “Efficient in-network moving object tracking in wireless sensor networks,” *Mobile Computing, IEEE Transactions on*, vol. 5, no. 8, pp. 1044 –1056, Aug. 2006.

[29] W. Zhang and G. Cao, “Optimizing tree reconfiguration for mobile target tracking in sensor networks,” in *INFOCOM 2004*, vol. 4, March 2004, pp. 2434 – 2445 vol.4.

[30] E. Olule, G. Wang, M. Guo, and M. Dong, “Rare: An energy-efficient target tracking protocol for wireless sensor networks,” in *ICPPW 2007*, Sept. 2007, p. 76.

[31] M. Wälchli, P. Skoczylas, M. Meer, and T. Braun, “Distributed event localization and tracking with wireless sensors,” ser. WWIC ’07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 247–258.

[32] Y. Xu, J. Winter, and W. chien Lee, “Prediction-based strategies for energy saving in object tracking sensor networks,” in *MDM’04 IEEE*, 2004, pp. 346–357.

[33] ——, “Dual prediction-based reporting for object tracking sensor networks,” in *MOBIQUITOUS 2004*, 2004, pp. 154–163.

[34] Y.-C. Tseng, S.-P. Kuo, H.-W. Lee, and C.-F. Huang, “Location tracking in a wireless sensor network by mobile agents and its data fusion strategies,” *The Computer Journal*, vol. 47, no. 4, pp. 448–460, 2004.

[35] S. Goel and T. Imielinski, “Prediction-based monitoring in sensor networks: taking lessons from mpeg,” *SIGCOMM Comput. Commun. Rev.*, vol. 31, pp. 82–98, October 2001.

[36] M. Chu, H. Haussecker, and F. Zhao, “Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks,” *International Journal of High Performance Computing Applications*, vol. 16, no. 3, pp. 293–313, 2002.

[37] X. ji, H. Zha, J. Metzner, and G. Kesidis, “Dynamic cluster structure for object detection and tracking in wireless ad-hoc sensor networks,” in *Communications, 2004 IEEE International Conference on*, vol. 7, June 2004, pp. 3807 – 3811 Vol.7.

[38] J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao, “Distributed group management for track initiation and maintenance in target localization applications.” in *IPSN’03*, 2003, pp. 113–128.

[39] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *Wireless Communications, IEEE Transactions on*, vol. 1, no. 4, pp. 660 – 670, Oct 2002.

[40] L. Fan, Z. Wang, and H. Wang, “Multi-target cell tracking based on classic kinetics,” ser. IWCMC ’09. ACM, 2009, pp. 1340–1344.

[41] M. Ding and X. Cheng, “Fault tolerant target tracking in sensor networks,” ser. MobiHoc ’09. ACM, 2009, pp. 125–134.

[42] O. Younis and S. Fahmy, “Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach,” in *INFOCOM 2004*, vol. 1, March 2004.

[43] F. Zhao, J. Shin, and J. Reich, “Information-driven dynamic sensor collaboration,” *Signal Processing Magazine, IEEE*, vol. 19, no. 2, pp. 61 –72, Mar 2002.

[44] H. Kung and D. Vlah, “Efficient location tracking using sensor networks,” in *IEEE WCNC*, vol. 3, March 2003, pp. 1954 –1961 vol.3.

[45] H. Yang and B. Sikdar, “A protocol for tracking mobile targets using sensor networks,” in *Sensor Network Protocols and Applications, 2003*, May 2003, pp. 71 – 81.

[46] W.-R. Chang, H.-T. Lin, and Z.-Z. Cheng, “Coda: A continuous object detection and tracking algorithm for wireless ad hoc sensor networks,” in *CCNC 2008*, Jan. 2008, pp. 168 –174.

[47] W.-P. Chen, J. Hou, and L. Sha, “Dynamic clustering for acoustic target tracking in wireless sensor networks,” *Mobile Computing, IEEE Transactions on*, vol. 3, no. 3, pp. 258 – 271, July-Aug. 2004.

[48] Z. Wang, H. Li, X. Shen, X. Sun, and Z. Wang, “Tracking and predicting moving targets in hierarchical sensor networks,” in *ICNSC 2008*, April 2008, pp. 1169 –1173.

[49] R. Brooks, P. Ramanathan, and A. Sayeed, “Distributed target classification and tracking in sensor networks,” *IEEE*, vol. 91, no. 8, pp. 1163 – 1171, Aug. 2003.

[50] L. Song and D. Hatzinakos, “A cross-layer architecture of wireless sensor networks for target tracking,” *IEEE/ACM Trans. Netw.*, vol. 15, pp. 145–158, February 2007.

[51] S. Sharma, S. Deshpande, and K. Sivalingam, “Alpha-beta filter based target tracking in clustered wireless sensor networks,” in *COMSNETS, 2011*, Jan. 2011, pp. 1 –4.

[52] J. Krumm and E. Horvitz, “Locadio: inferring motion and location from wi-fi signal strengths,” in *MOBIQUITOUS 2004*, Aug. 2004, pp. 4 – 13.

[53] T. S. Ito, M. and Y. Kosuge, “Tracking a three-dimensional moving target with distributed passive sensors using extended kalman filter,” *Electronics and Communications in Japan (Part I: Communications)*, vol. 84, pp. 74–85, March 2001.

[54] M. Arslan, T. Onel, and C. Ersoy, “Effect of 3d topographical surfaces for the performance evaluation of wireless sensor networks,” in *PIMRC 2008*, Sept. 2008, pp. 1 –6.

[55] F. Thomas and L. Ros, “Revisiting trilateration for robot localization,” *Robotics, IEEE Transactions on*, vol. 21, no. 1, pp. 93 – 101, Feb. 2005.

[56] D. Manolakis, “Efficient solution and performance analysis of 3-d position estimation by trilateration,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 32, no. 4, pp. 1239 –1248, Oct 1996.

[57] B. T. Fang, “Trilateration and extension to global positioning system navigation,” *Journal of Guidance Control Dynamics*, vol. 9, pp. 715–717, december 1986.

[58] (2012) Microstation inertial sensors. [Online]. Available: http://www.microstrain.com/inertial/3dm-gx3-35

[59] (2012) Sensortips - inclinometer sensors. [Online]. Available: http://www.sensortips.com/category/zlevel/inclinometer/

[60] G. M. Siouris, *An Engineering Approach to Optimal Control and Estimation Theory*. John Wiley & Sons, Inc., 1996.

[61] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[62] E. Brookner, *Tracking and Kalman Filtering Made Easy*. Wiley-Interscience, Apr. 1998.

[63] Y.-S. Chiou, C.-L. Wang, and S.-C. Yeh, “Reduced-complexity scheme using alpha-beta filtering for location tracking,” *Communications, IET*, vol. 5, no. 13, pp. 1806 –1813, 5 2011.