

Decentralized UAV Tracking with Networked Radar Systems

Michael Eyler,^{*} Brady Anderson,[†] Cameron K. Peterson[‡], Tim McLain[§], and Karl F. Warnick[¶]
Brigham Young University, Provo, UT, 84602

I. Introduction

As traffic from unmanned aircraft increases, so does the need for infrastructure capable of detecting and tracking them. Research investigating both decentralized tracking algorithms and software architectures is critical to mitigate the struggles of centralized systems to collect, process, and disperse information in a timely manner. In this paper we demonstrate a simulated air traffic system that tracks targets across a large geographical area using a decentralized radar network.

Grime and Durrant-Whyte demonstrated a decentralized data fusion algorithm that combined sensors in a way that has greater network flexibility and robustness than a centralized system, yet produced similar results [1]. Aldosari and Moura discussed the ability of sensors constrained by power, size, and computing capacity to work together and how to take those constraints into proper consideration when designing networks for their use [2]. Xie et al. calculated an improved lower-bound estimate for both centralized and decentralized systems and also developed a strategy for decentralized radar network design while observing resource constraints [3].

Several studies, [4–12] have researched decentralized target tracking and state estimation algorithms. The decentralized hybrid information fusion (DHIF) algorithm, which is used in this paper, was first introduced in [4] and later improved in [5–7, 13]. It is built upon the Kalman information filter and fuses measurements from sensors across a decentralized network while maintaining filter consistency.

In [5, 6], Wang and Ren adjusted DHIF to account for time-varying process models and network topology, simulated it, and investigated the conditions necessary for convergence. They continued in [7] by including nonlinear properties in both the target models and estimator models. In [8], the algorithm was extended even further using the unscented transform to increase the algorithm's robustness against dropped messages or poor network strength. Indoor experiments, tested with data from a small wheeled robot, validated the nonlinear algorithm. However, in all the works by these authors, none of the simulations or experiments included background clutter or tracked multiple targets.

Several other authors have also introduced decentralized algorithms. Bishop et al. [9] introduced a nonlinear decentralized target-tracking algorithm and demonstrated the boundedness on the error of the estimate. This was done arithmetically and using a simulation of a radar network. However, they only included a single target and had no clutter in the return. Li et al. [10] demonstrated a distributed observer that guarantees asymptotic stability of the error set (including input-to-state stability) using both analytical and numerical validation.

References [11] and [12] compared their decentralized algorithms with DHIF. Wu et al. [11] introduced a decentralized algorithm for continuous-time-based systems and compared it with DHIF via simulations. However they required that all sensors view the target. Deshmukh et al. [12] proposed a distributed algorithm using multiple state models to estimate the target state and optimal consensus to process information. They then contrasted their algorithm with DHIF, and provided an example of tracking a single aircraft in the absence of clutter.

This work creates a decentralized filter which combines measurements to track multiple targets across a radar network, something that was not done by any of the previous authors. It does this by using the linear version of DHIF, the recursive random sample consensus algorithm (RRANSAC), a variant of the random sample consensus algorithm (RANSAC), and the sequential covariance intersection algorithm (SCI). RRANSAC, which was developed in [14], builds on RANSAC by recursively searching for an unknown number of valid targets in the midst of clutter and noise. It is used in this paper to separate background clutter from true targets and initialize the track estimates for DHIF.

SCI, first introduced in [15], was developed as a means to simplify the process of determining the proper weights to apply during information fusion. It operates by splitting up the multi-dimensional optimization problem into a string of one-dimensional optimization problems that can be calculated iteratively. We use this to help with the information

^{*}Student, Electrical and Computer Engineering Department.

[†]Student, Mechanical Engineering Department.

[‡]Assistant Professor, Electrical and Computer Engineering Department, Senior Member.

[§]Professor, Mechanical Engineering Department.

[¶]Assistant Professor, Electrical and Computer Engineering Department

fusion step by determining the proper weights for each estimate as soon as that information is available, instead of waiting for a designated update interval.

This paper leverages prior work by the authors[16] which first applied RRANSAC to a network of low cost, size, weight, and power (low CSWAP) radars. The results demonstrated the utility of connecting multiple radar systems to cover a large area, but was limited to a centralized solution that ultimately restricted its scalability.

By using DHIF in conjunction with these tools, this work demonstrates the ability of a decentralized radar system to identify multiple targets in the midst of background clutter, estimate their states, and track them across multiple radar fields of view. Numerical simulations demonstrate the utility of this approach.

The remainder of this work continues as follows. Section II presents the methods used in the decentralized air traffic tracking simulation. This includes how radar detections are filtered for targets and tracked across the network. Section III explores the simulation results. Lastly, Section IV discusses the conclusions and future directions.

II. Methods

In this section we will describe the components of the decentralized local air traffic tracking system. In prior work, the authors built a network of radar ground stations that collected data and processed the information centrally. For context, this work is briefly summarized in Section II.A. In Section II.B, we will explain the structure of the new decentralized system used in this work. Finally, in Section II.C we describe our implementation of the DHIF filter and how it is combined with RRANSAC to track air targets.

A. Centralized Local Air Traffic Network System

In [16] a network of radar ground stations were used to track unmanned air vehicles across multiple radar fields of view. Similar results were obtained with simulation. Each ground station was equipped with a low CSWAP radar that used a phased array antenna to capture data. The radar detections were communicated to and processed at a centralized ground station that used the RRANSAC algorithm to remove noise, detect valid targets, and estimate their states.

A key difference between this work and the previous work of the authors is in the implementation of RRANSAC. In prior work, target information was collected and run through RRANSAC at a regular interval. In this work, the algorithm operates locally on data received from the radar as soon as it is made available to the function. Another benefit of using RRANSAC locally is that it reduces the information being processed by the algorithm to what is seen by a single radar instead of the entire network, making it more computationally tractable. A brief description of how RRANSAC operates on this information is described next.

The RRANSAC algorithm compares current information alongside previous measurements in a sliding window to filter out noise and find motion models that meet certain criteria. A good model is characterised by user-defined parameters at the initialization of the filter, such as the percentage of inliers per model, the number of consecutive detections before a model is reported, and the number of missed detections before a track is considered obsolete. RRANSAC compares different track models to the data recursively until appropriate models are discovered. These models are then updated through time as more data is received, until the target being tracked leaves the field of view. More information about the specifics of RRANSAC and how it has been applied to radar systems can be found in [14, 16].

B. Decentralized Local Air Traffic Network System

For the simulation in this work, the targets were generated in the operational area with a constant linear velocity. Randomly generated background clutter, along with the target locations (if they were in the calculated field of view), were passed to the radar's UDP receiver over the network. The exact target locations were altered slightly by adding truncated, normally distributed Gaussian noise to the measurement, to better match measurements coming from physical radars. Each range measurement was divided into 0.6 m increments, while the zenith and azimuth measurements each received noise from a normally distributed Gaussian with a sigma of 1 degree that were truncated at 4 degrees.

Once received, the information was converted from local range, zenith, and azimuth coordinates to a global north, east, down coordinate frame and sent through a local instance of RRANSAC to remove noise, identify tracks, and estimate the states of those tracks. These tracks were then passed to a local instantiation of DHIF (known as an agent), along with the original measurement, to be initialized or updated. At that point, all of the recently updated tracks in the agent's repository were broadcast to neighboring agents who were subscribed to that node. They would then use the information to initialize or update their own track database. Figure 1 shows the architecture of the physical system, which is used in the simulation for this work. The only difference here being that the measurements are generated within

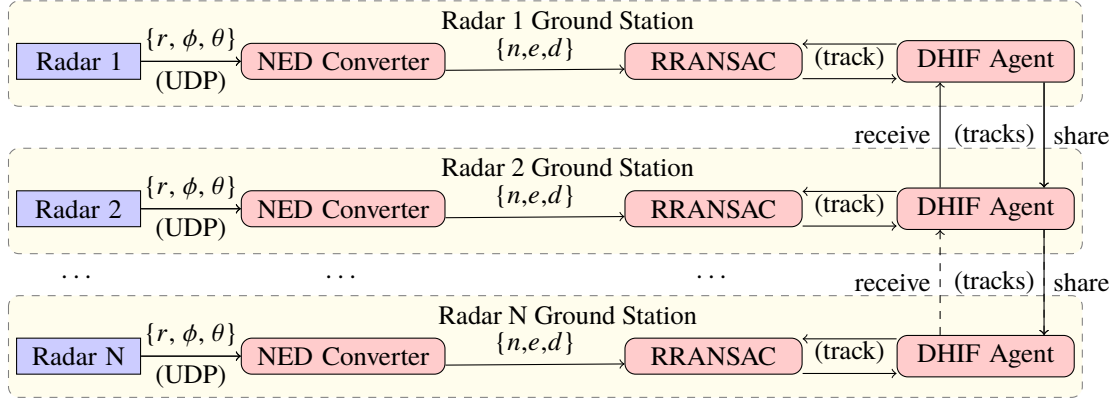


Fig. 1 System Architecture for the simulated local air traffic monitoring system demonstrated in this work. Any clutter or other noise, along with targets calculated to be within the field of view of a given radar, are sent over UDP to the local ground station as measurements containing range, r , zenith, θ , and azimuth, ϕ . The detection coordinates are then converted to a global NED frame and the north, n , east, e , and down, d measurements are processed with RRANSAC before the tracks that are identified are sent to the local DHIF agent. Those tracks are then processed and shared across the decentralized network where other agents can use the data.

the simulation instead of being provided by a physical radar.

C. Decentralized Hybrid Information Filter

Though this work is based upon the linear DHIF algorithm as seen in[4], it has been modified slightly to be able to operate with multiple targets in a cluttered background with updates occurring asynchronously. DHIF is a hybrid of two information fusion steps which operate across a decentralized network of sensors. It uses the Kalman information filter, which operates as a Kalman filter, except the state covariance and estimate have been inverted and multiplied by the inverse of the estimate to simplify the measurement update step. DHIF uses this simplification to its advantage as seen in the following steps.

After a sensor initializes a DHIF agent (agents collect, update, and send/receive track information) the agent can begin processing information. At time t , a track, i , is initialized with a state estimate, $\hat{x}_{i,t|t-1}$, and a covariance bound, $P_{i,t|t-1}$. In the case that these previous estimates are not available, the inverse of the estimate can be initialized with $[0]$ as discussed in[4].

First, the state information vector, $\xi_{i,t}$, and information matrix, $\Xi_{i,t}$, are calculated as follows:

$$\xi_{i,t} = P_{i,t|t-1}^{-1} \hat{x}_{i,t|t-1} \quad \Xi_{i,t} = P_{i,t|t-1}^{-1} \quad (1)$$

Then, using the measurement, $z_{i,t}$, and its covariance, $R_{i,t}$, the information equivalent of the measurement matrix and its co-variance are calculated,

$$y_{i,t} = H^T R_{i,t}^{-1} z_{i,t} \quad S_{i,t} = H^T R_{i,t}^{-1} H, \quad (2)$$

where H is observation matrix of the radar measurements. Next, the vectors and matrices for all tracks 1 through n are packaged together as a track message,

$$T_t = \{\{\xi_{1,t}, \Xi_{1,t}, y_{1,t}, S_{1,t}\}, \dots, \{\xi_{n,t}, \Xi_{n,t}, y_{n,t}, S_{n,t}\}\}, \quad (3)$$

and broadcast to other agents that have subscribed. (These agents are denoted as neighbors.)

When track information is received from neighbors, the information vectors and matrices are appropriately weighted and combined with an existing track or a new track is initialized like the following (where $j \in J$ represent all of the available measurements for track i at time t),

$$\bar{\Xi}_{i,t} = \sum_j w_j \Xi_{(i,t),j} \quad \bar{\xi}_{i,t} = \sum_j w_j \xi_{(i,t),j}. \quad (4)$$

Likewise, the measurements in the messages are combined together as follows

$$\bar{S}_{i,t} = \sum_j^J S_{(i,t),j} \quad \bar{y}_{i,t} = \sum_j^J y_{(i,t),j}, \quad (5)$$

At that point, the state vector and covariance are extracted from the state information vector and information matrix as

$$P_{i,t|t} = (\bar{S}_{i,t} + \bar{\Xi}_{i,t})^{-1} \quad \hat{x}_{i,t|t} = P_{i,t|t}(\bar{\xi}_{i,t} + \bar{y}_{i,t}). \quad (6)$$

Finally, the tracks are propagated, and stored until the next update using the simple Kalman update

$$P_{i,t|t+1} = F P_{i,t|t} F^T + B Q B^T \quad \hat{x}_{i,t|t+1} = F \hat{x}_{i,t|t}, \quad (7)$$

where F is state transition matrix, and the matrices B and Q which model the process noise of the system and represent the noise covariance. Proofs and more details regarding this process can be found in [4]. (Although this works for any linear, time-invariant model, this study used a constant-jerk model to generate both F and Q . B was set to the identity.)

Our use of DHIF relies on RRANSAC to isolate the true measurements from the clutter. This combination filters out the noise from the radar units and also select the targets to track. Thus DHIF can operate on a system without requiring measurements to be noiseless or well ordered (i.e. the tracks are not required to come in the same order every time).

Extra steps were required alongside the original DHIF algorithm to track multiple targets in the midst of clutter. This includes three data association filters, utilizing universally unique IDs and Mahalanobis distances, as well as the sequential covariance intersection (SCI) algorithm [15] for determining proper track weights.

The first data association filter is used when integrating measurements from RRANSAC. Let a message be received by agent q from RRANSAC at time t be

$$M_t = \{\{\hat{x}_i, P_i, z_i, R_i\}, \dots, \{\hat{x}_n, P_n, z_n, R_n\}\}. \quad (8)$$

The Mahalanobis distance between the incoming track estimates and each of the local tracks, T_q , is then calculated and compared with a maximum allowable distance, d_m , which ensures that the correct measurement is paired with its estimate. The timestamps of the local and incoming tracks are then compared to determine if the track needs to be propagated forward (Equation (7)), and finally, the track is updated (Equations (1), (2), (4), and (5)). This process is seen in Algorithm 1.

If no match is found, a new track is initialized with a universally unique track ID. This new track requires the previous state estimate and covariance, $\hat{x}_{t|t-1}$ and $P_{t|t-1}$, to calculate the information vector and information matrix, ξ_t and Ξ_t . This is backed out from the current state estimate and covariance $\hat{x}_{t|t}$ and $P_{t|t}$ provided by RRANSAC. Inverting Equation (6) yields

$$P_{i,t|t-1} = (P_{i,t|t}^{-1} - S_{i,t})^{-1} \quad \hat{x}_{i,t|t-1} = P_{i,t|t}^{-1} \hat{x}_{i,t|t} - y_{i,t}. \quad (9)$$

The track ID generated is used for the second data association filter, when a message containing track information is received from agent r at time t , $T_{t,r}$. First, all the track IDs of the local agent q are compared with incoming track IDs to determine any matches. If the IDs match, the track estimates are propagated and updated appropriately. Otherwise, they go on to the next data association filter which again uses the Mahalanobis distance to compare local and incoming tracks. If the distance is small enough that a match can be confirmed, the tracks are propagated and updated just as before. Any remaining tracks at this point are initialized locally (with the track ID and estimate provided in the message). These three data association filters work together to prevent redundant tracks and ensure proper matches of measurements and estimates. This entire process is seen in Algorithm 2.

When combining estimates, the original DHIF algorithm calculated the estimate weights after it received all the measurements for a given target in that time step. These weights assisted DHIF in combining estimates while ensuring they were not redundant or overly confident. To prevent the sum of the estimates from becoming greater than the truth, the weights are bounded by 0 and 1 and the sum of the weights is equal to 1. However, DHIF did not account for the number of agents that observed the target to change over time, making it difficult to determine when all available measurements have been collected.

Algorithm 1: Agent q processing an RRANSAC measurement message, M_t .

```

1  Update the local agent's current time to  $t$ ;
2  if  $T_q = \emptyset$  then
3      Create track  $i \in M_t$  (Eqs. (9), (1), (2), (4), and (5));
4      add track:  $T_q \leftarrow i$ ;
5  end if
6  for tracks:  $j \in M_t \neq i$  do
7      Match  $\rightarrow$  False;
8      for tracks:  $k \in T_q$  do
9          if Mahalanobis( $\hat{x}_{j,t|t}, \hat{x}_{k,t|t-1}, P_{j,k|k}$ )  $< d_m$  then
10             if time( $k$ )  $< t$  then
11                 Propagate  $k$  (Eqs. (6) and (7));
12                 time( $k$ ) =  $t$ ;
13             end if
14             Update  $k$  (Eqs. (1), (2), (4), and (5));
15             Match  $\rightarrow$  True;
16             break;
17         end if
18     end for
19     if Match = False then
20         Create track  $j$  (Eqs. (9), (1), (2), (4), and (5));
21         add it:  $T_q \leftarrow j$ ;
22     end if
23 end for
24  $T_{t,q} \leftarrow$  tracks  $i \in T_q \mid \text{time}(i) = t$ ;
25 Share  $T_{t,q}$  with neighboring agents.

```

Algorithm 2: Agent q processing a DHIF Track message from agent r , $T_{t,r}$.

```
1 Update the local agent's current time to  $t$ ;  
2 if  $T_q = \emptyset$  then  
3   Create track  $i \in T_{t,r}$  (Eqs. (4) and (5));  
4   add track:  $T_q \leftarrow i$ ;  
5 end if  
6 for tracks:  $j \in T_{t,r} \neq i$  do  
7   for tracks:  $k \in T_q$  do  
8     Match  $\rightarrow$  False;  
9     if  $ID(j) = ID(k)$  or  
10    Mahalanobis( $\hat{x}_{j,t|t}, \hat{x}_{k,t|t-1}, P_{j,k|k}$ )  $< d_m$  then  
11      if  $\text{time}(k) < t$  then  
12        Propagate  $k$  (Eqs. (6), (7), and (1));  
13         $\text{time}(k) = t$ ;  
14      end if  
15      Calculate the weights using SCI (Eq. (10));  
16      Update  $k$  (Eqs. (4) and (5));  
17      Match  $\rightarrow$  True;  
18      break;  
19    end if  
20  end for  
21  if Match = False then  
22    Create track  $j$  (Eqs. (4) and (5));  
23    add it:  $T_q \leftarrow j$ ;  
24  end if  
25 end for  
26  $T_{t,q} \leftarrow$  tracks  $i \in T_q \mid \text{time}(i) = t$ ;  
27 Share  $T_{t,q}$  with neighboring agents.
```

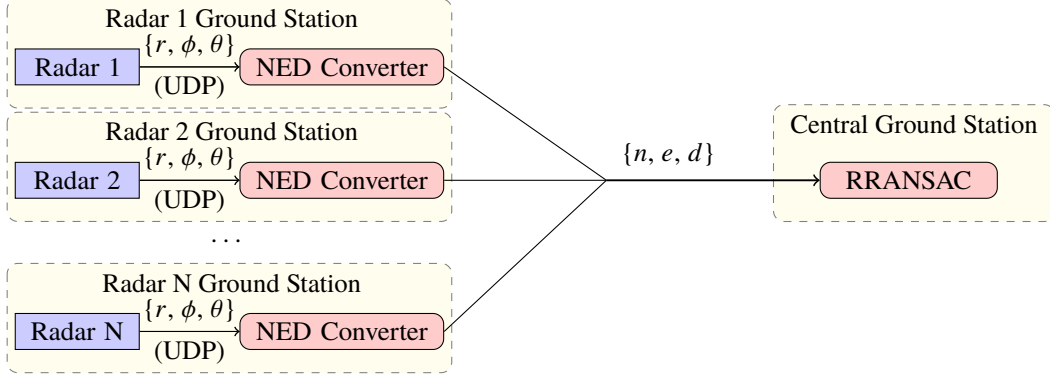


Fig. 2 System Architecture for the physical centralized local air traffic monitoring system, which is simulated in this work. The clutter and detection measurements (range r , zenith θ , and azimuth ϕ) from each radar are sent over UDP to a local ground station. After the information has been received and converted to a global NED frame, the detections are sent to the central ground station which uses RRANSAC for complete track identification and state estimation.

Instead, this work uses the SCI algorithm, which allows us to calculate and apply the weights, w_q and w_r , to the track estimates, Ξ_q and Ξ_r , as they become available. The SCI algorithm adapted from [15] to apply to the DHIF filter is

$$\begin{aligned}
 [w_q, w_r] = \arg \min_{w_{q,r}} & \quad \left(\text{trace} (w_q \Xi_q + w_r \Xi_r)^{-1} \right) \\
 \text{s.t.} & \quad w_q + w_r = 1 \\
 & \quad 0 \leq w_{q,r} \leq 1.
 \end{aligned} \tag{10}$$

III. Results

The scenario used in this simulation involves many ground-based radar stations dispersed in an urban environment as seen in Figure 3. The radars are laid out to ensure complete coverage over a roughly 800 m x 700 m area and the DHIF filter is tested in its ability to detect small UAVs and share tracks across the entire decentralized network. To model the noise in the radar's measurements, each range measurement was divided into 0.6 m bins, while the zenith and azimuth measurements each received noise from a normally distributed Gaussian with a sigma of 1 degree that was truncated at 4 degrees. Each radar has a 130 m sensing radius and communicates with other radar systems that lie within 300 m of its location at a rate of 1-2 Hz. These include the radars for which they share overlapping regions. To provide a baseline, the same scenarios are tested using a centralized network of radar stations. The centralized architecture matching the previous work [16] can be seen in Figure 2. We used the same setting for RRANSAC as in our previous work [16] with the exception of $\mathcal{M}$ (the number of models RRANSAC should keep at any time), which we increased to 30 because we had less false returns with this setup and were tracking more targets.

For this work, both architectures used the same target paths to provide a valid comparison between the results. The pathways selected (seen in Figure 3) were chosen to cover a large distance, demonstrate different constant-acceleration paths, and pass through multiple radar fields of view. The results of both architectures can be seen in the Figures 4-7, and are described in the subsequent paragraphs.

Figures 4 and 5 represent the results of the centralized network. Figure 4 shows the absolute error for each target track over the entire test. Each red star represents the error from truth of the measurement that the radar returned, the solid green line shows the estimate error for the target track, and the black dashed line is the 3-sigma bound for each estimate's covariance. The initial variance on each track raises and dips slightly, this is an artifact of initialization process inside the local RRANSAC instance. It's important to note that the changes in covariance reflect the radar's measurement covariance which is time varying. This is because the noise on measurements is dependent on the range and azimuth values. At further range, the Cartesian values of the measurement covariance matrix ($R_{i,t}$) increase. Figure 5 is a close up of Figure 4 and uses the same notation. From this figure it is easier to see that RRANSAC provides an accurate estimate of the target given the noisy measurements. Due to limitations in the centralized system, the results

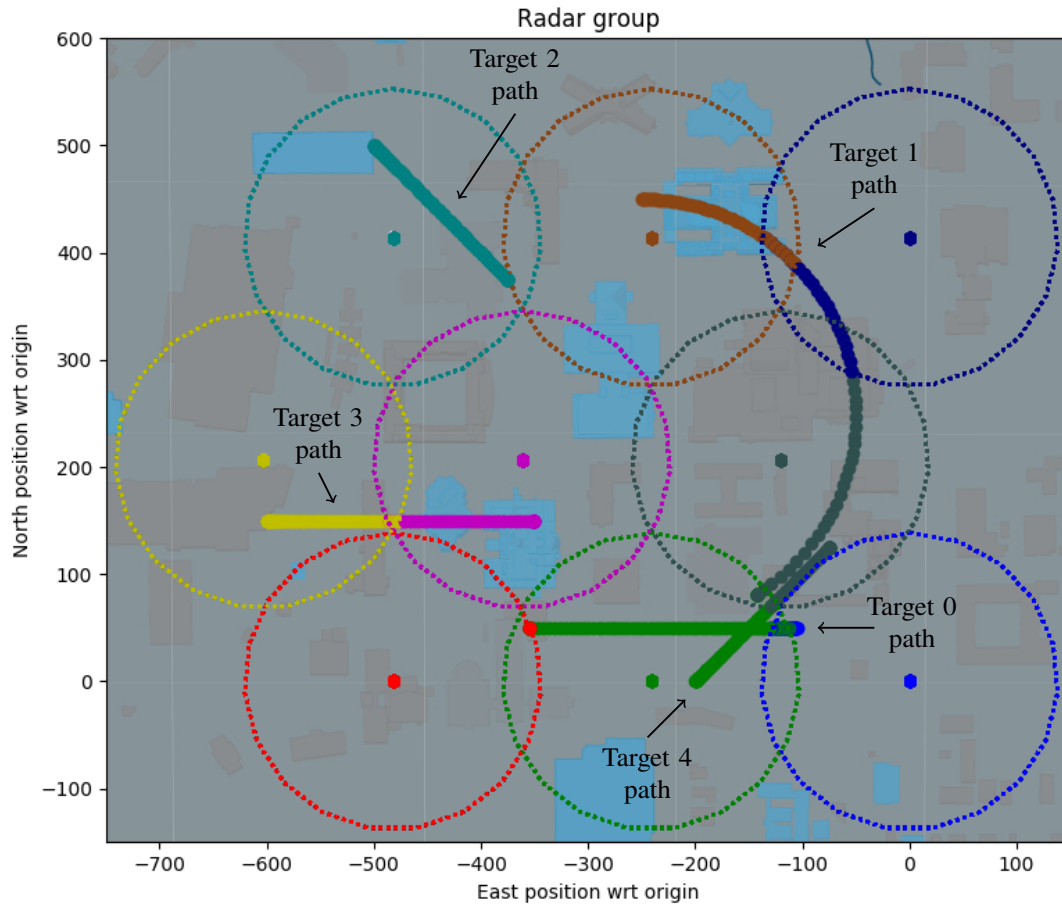


Fig. 3 These constant acceleration pathways were used in both the centralized and decentralized simulations to test the capabilities of the systems. The paths are shown here without added noise to the measurements or clutter in the background. The color changes denote which radar was observing the target at that location. In overlapping regions the target is observed by both radars, but the path (color) of the second radar's detections is obscured by the first.

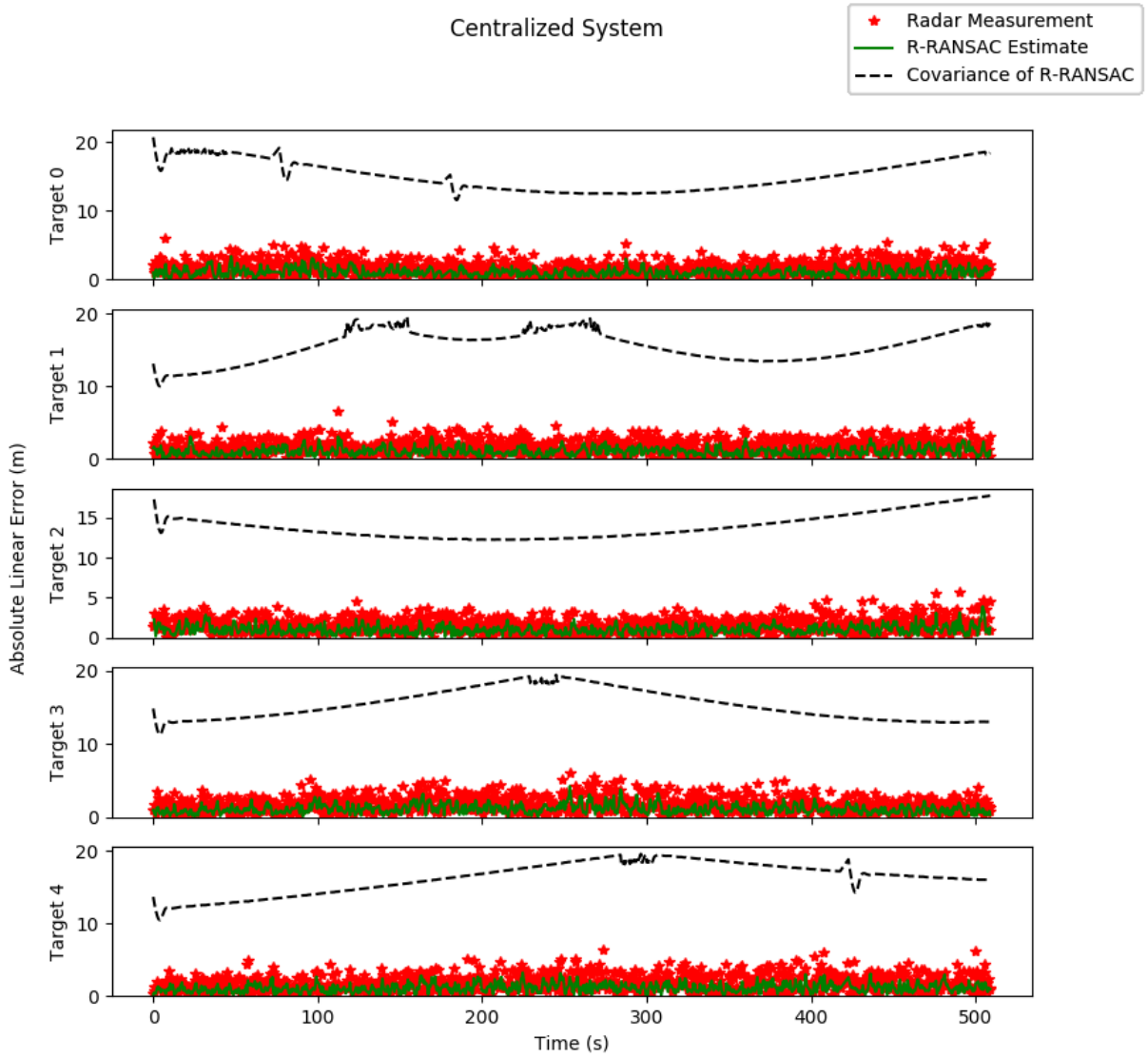


Fig. 4 The absolute error (m) of the estimate per target (green solid line) provided by RRANSAC compared to the truth data. The red stars represent radar measurements and the black dashed line is the approximate 3-sigma bound for each estimate.

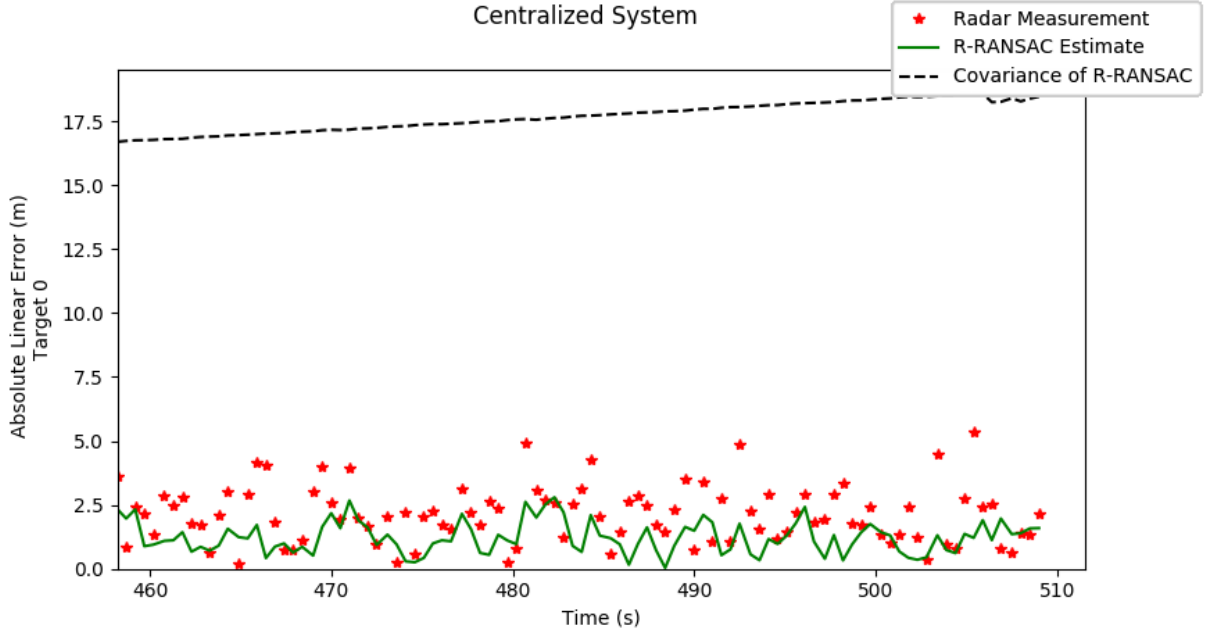


Fig. 5 A close up of the centralized RRANSAC estimate for target 0 between the times of $t = [460, 510]$. This region indicates that the target is moving away from the radar system causing its measurement covariance to rise.

presented here reflect the systems ability to track multiple targets only in the absence of clutter, something that the decentralized system did not struggle with.

Figures 6 and 7 both show the results of the decentralized network. Similar to Figure 4, Figure 6 provides the error associated with each target for the duration of the test. The red stars also represent the absolute error of radar measurements relative to the target's real position, but what appear to be the other two lines are the estimate errors of all of the agents (solid lines are estimate error, dashed lines are the 3-sigma bound). Occasionally the lines do diverge slightly, this is likely due to issues in connectivity or timing. Something that could completely derail a centralized system if it occurred at the wrong time or place. Finally, it is also significant to point out the dramatic decrease in covariance error whenever a target is seen by two radars. This can be seen in Figure 7.

From these plots we see that the decentralized implementation accurately tracks and estimates the targets with an error that is comparable to a centralized implementation. This is possible even when the targets pass beyond the radar's field of view or its personal range of direct communication. DHIF provides this increased flexibility and robustness when tracking across the network, which, in turn enables better support for the infrastructure needs of unmanned aircraft in the future.

IV. Conclusion

This work presented a decentralized tracking filter capable of fusing information across a radar network. The DHIF filter provided a way to share information across the entire network without the need for a centralized ground station. Simulations showed that the DHIF filter will effectively track multiple targets in a cluttered environment across a decentralized radar network.

The simulations used here were developed to model the physical radar network structure and lay the foundation for future tests involving hardware results. Future work will also include the use of non-linear state estimators and filters in both DHIF and RRANSAC. As well as more sophisticated data association filters.

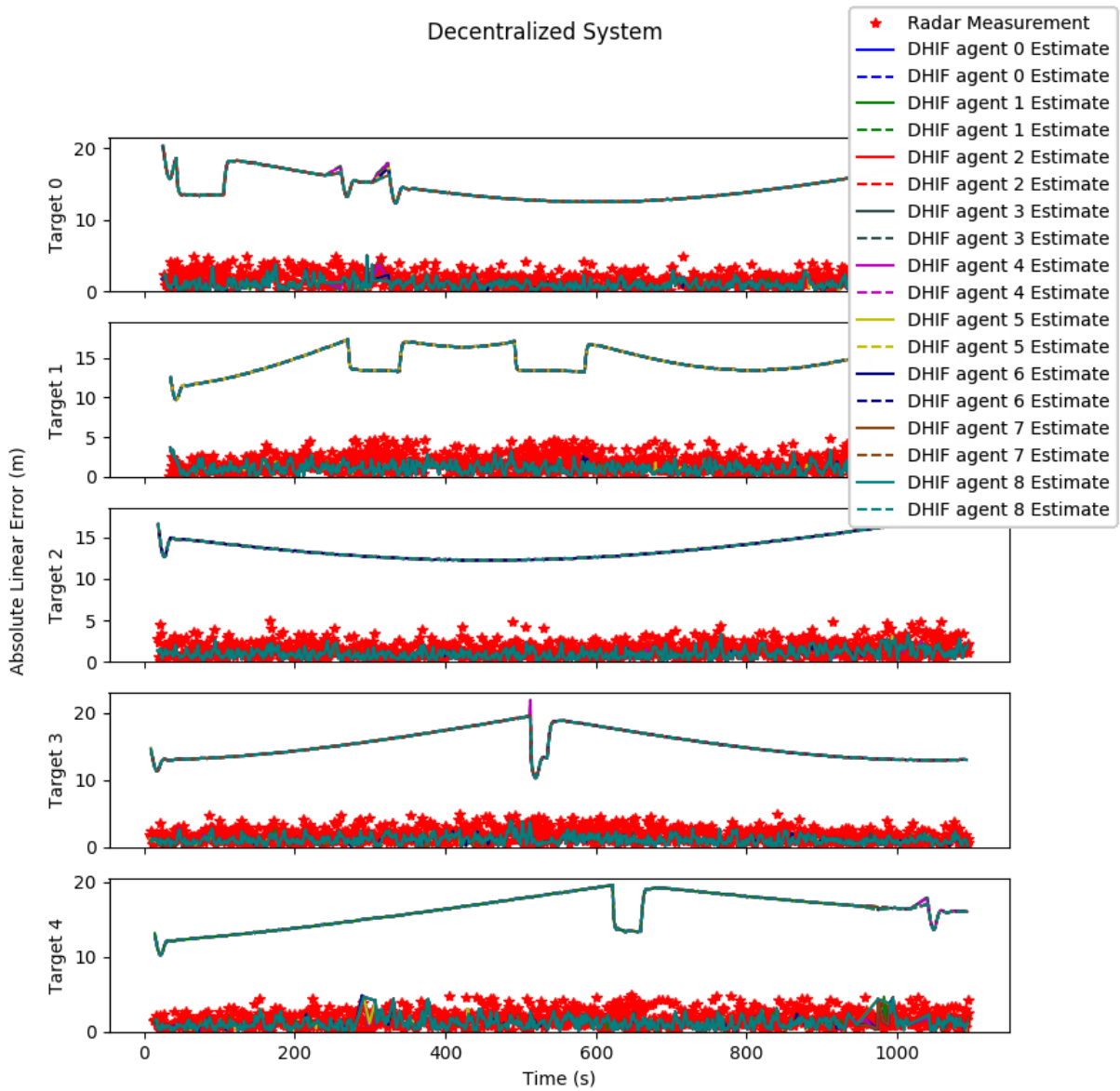


Fig. 6 This plot summarizes the absolute error for each track estimate (solid lines) and covariance (dashed lines). Radar measurements are represented by red stars. The rapid spike followed by a dip in the covariance occur when a track is momentarily lost and re initialized. The longer, flatter dips in the covariance represent the combination of two radar measurements.

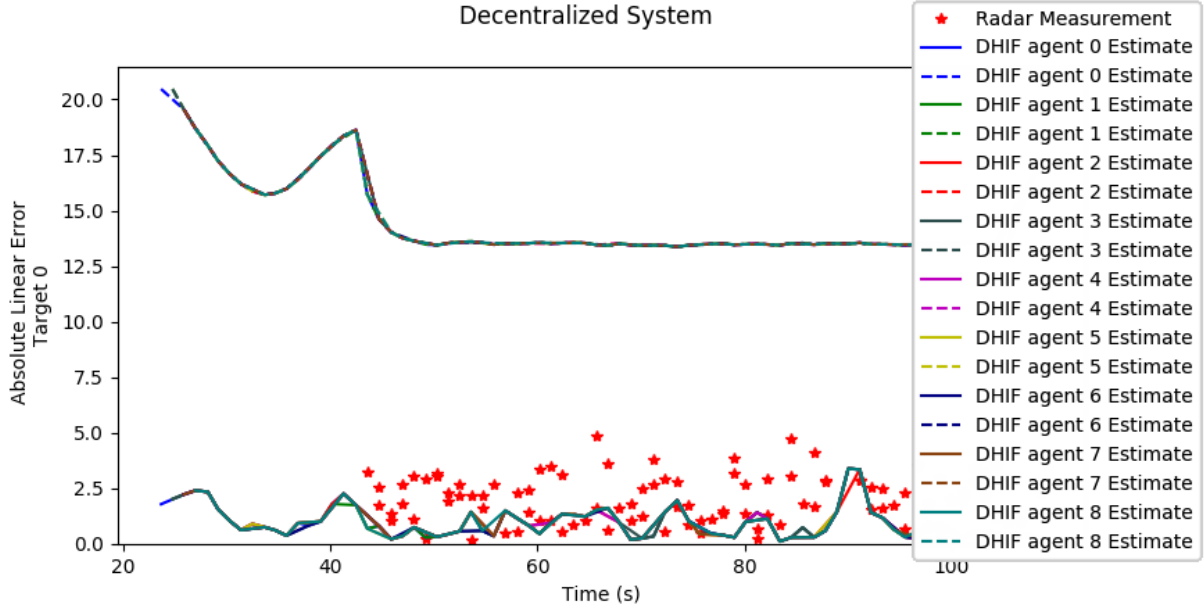


Fig. 7 A close up of the estimate for target 0 from all agents between the times of $t = [20, 100]$. After the track has been initialized, there is a drop followed by a flat 3-sigma covariance bound. This region indicates that the target is being detected by two radars at the same time.

Acknowledgments

This work was supported by the National Science Foundation under Grant No. 1727010. This work is also partially supported by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation Industry/University Cooperative Research Center (IUCRC) under NSF award No. IIP-1161036 along with significant contributions from C-UAS industry members.

References

- [1] Grime, S., and Durrant-Whyte, H. F., "Data fusion in decentralized sensor networks," *Control engineering practice*, Vol. 2, No. 5, 1994, pp. 849–863.
- [2] Aldosari, S. A., and Moura, J. M., "Detection in decentralized sensor networks," *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 2, IEEE, 2004, pp. ii–277.
- [3] Xie, M., Yi, W., Kirubarajan, T., and Kong, L., "Joint node selection and power allocation strategy for multitarget tracking in decentralized radar networks," *IEEE Transactions on Signal Processing*, Vol. 66, No. 3, 2017, pp. 729–743.
- [4] Wang, S., and Ren, W., "On the consistency and confidence of distributed dynamic state estimation in wireless sensor networks," *2015 54th IEEE Conference on Decision and Control (CDC)*, IEEE, 2015, pp. 3069–3074.
- [5] Wang, S., and Ren, W., "On the convergence of distributed estimation of LTV dynamic system with switching directed topologies and time-varying sensing models," *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 5437–5442.
- [6] Wang, S., and Ren, W., "On the convergence conditions of distributed dynamic state estimation using sensor networks: A unified framework," *IEEE Transactions on Control Systems Technology*, Vol. 26, No. 4, 2017, pp. 1300–1316.
- [7] Wang, S., and Ren, W., "Fully distributed nonlinear state estimation using sensor networks," *2017 American Control Conference (ACC)*, IEEE, 2017, pp. 2562–2567.
- [8] Wang, S., Lyu, Y., and Ren, W., "Unscented-Transformation-Based Distributed Nonlinear State Estimation: Algorithm, Analysis, and Experiments," *IEEE Transactions on Control Systems Technology*, Vol. 27, No. 5, 2018, pp. 2016–2029.
- [9] Bishop, A. N., Pathirana, P. N., and Savkin, A. V., "Decentralized and robust target tracking with sensor networks," *IFAC Proceedings Volumes*, Vol. 41, No. 2, 2008, pp. 14969–14975.

- [10] Li, Y., Phillips, S., and Sanfelice, R. G., "On distributed observers for linear time-invariant systems under intermittent information constraints," *IFAC-PapersOnLine*, Vol. 49, No. 18, 2016, pp. 654–659.
- [11] Wu, J., Elser, A., Zeng, S., and Allgöwer, F., "Consensus-based distributed Kalman-Bucy filter for continuous-time systems," *IFAC-PapersOnLine*, Vol. 49, No. 22, 2016, pp. 321–326.
- [12] Deshmukh, R., Thapliyal, O., Kwon, C., and Hwang, I., "Distributed state estimation for a stochastic linear hybrid system over a sensor network," *IET Control Theory & Applications*, Vol. 12, No. 10, 2018, pp. 1456–1464.
- [13] Wang, S., "Estimation in Networked Systems: Power Grid Security and Distributed Hybrid Information Fusion Algorithm," Ph.D. thesis, UC Riverside, 2017.
- [14] Niedfeldt, P. C., and Beard, R. W., "Recursive RANSAC: multiple signal estimation with outliers," *9th IFAC Symposium on Nonlinear Control Systems*, Vol. 16, 2013, pp. 45–50.
- [15] Deng, Z., Zhang, P., Qi, W., Liu, J., and Gao, Y., "Sequential covariance intersection fusion Kalman filter," *Information Sciences*, Vol. 189, 2012, pp. 293–309.
- [16] Anderson, B., Ellingson, J., Eyler, M., Buck, D., Peterson, C. K., McLain, T., and Warnick, K. F., "Networked Radar Systems for Cooperative Tracking of UAVs," *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2019, pp. 909–915.