

Network Centric MFA Tracking Architecture Based on Soft-Level Data Association

Roman Novoselov, Sabino Gadaleta, and Aubrey Poore

Numerica Corporation, P.O. Box 271246, Fort Collins, CO 80527

ABSTRACT

This work presents a new network centric architecture for multiple frame assignment (MFA) tracking. The architecture improves on earlier network tracking schemes by allowing trackers to broadcast decisions about their local soft-level associations, via the Soft Associated Measurement Reports (SAMRs). The SAMR may be followed by an “Oops” message, if the soft association was incorrect and must be revoked. We show, however, that such revocations are very rare in most scenarios. This paper discusses the implementation of the new algorithm and presents simulation results. Considerable improvements in the consistency of the air picture are demonstrated, owing to the the reduced latency in transmission of measurement-to-track associations. The earlier network architectures, namely, the Centralized MFA, the Replicated Centralized MFA, and the Network MFA on Local and All Data, are also discussed in this work, as they form the foundation for the “Oops” algorithm.

Keywords: Multiple hypothesis tracking, multiple frame assignment, distributed tracking architecture.

1. INTRODUCTION

Fusion of data from sensors distributed across multiple platforms has the potential to significantly enhance tracking performance. Specific benefits include larger coverage volume, improved state estimation, and greater discrimination accuracy. In such distributed environments, tracking algorithms function within a predefined communication scheme or architecture. Each architecture must address a host of network-related topics, including (1) distributed data association and estimation; (2) maintenance of a single integrate air picture (SIAP); (3) management of communication loading (4) sensor biases as well as location and registration errors (sometimes called gridlock); (5) pedigree problems; and, (6) out-of-order, latent, and missing data due to both sensor and communication problems. These topics as well as others are discussed in the article by Moore and Blair¹ and the book by Blackman and Popoli.²

The network architecture described in¹ provides a consistent air picture across multiple platforms and limits the communications loads to within a practical limit. This architecture is, however, designed with single frame data association in mind. The multiple frame methods, such as the multiple hypothesis tracking (MHT) and multiple frame assignment (MFA) offer substantially improved tracking and discrimination performance compared to the current single frame process. Thus, an important objective is to develop network-centric tracking architectures specific to the multiple frame algorithms. These architectures must insure track quality, maintain SIAP, and minimize the communication loads.

Over the past several years Numerica has developed a set of distributed network architectures in support of multiple frame tracking. These architectures are discussed in^{3,4} and are briefly outlined in this paper. In general, these algorithms provide accuracy similar to that of a centralized tracker, while maintaining SIAP across the network of platforms, and managing the communication loading. One architecture, Network MFA on All Data, is particularly well suited for scenarios with non-overlapping sensor coverage, since both local and remote data is used in assigning measurements to tracks. On the other hand, Network MFA on Local Data has lower communication load and achieves better SIAP when sensor coverage volumes do overlap.

Both of these architectures, however, suffer from a delay in sending out the data association decisions until measurement become firmly assigned to the network tracks. These firm assignments are made on the back pane of the moving window (which houses the multiple frames used for data association), and each frame must be propagated to that level. During

Further author information: (Send correspondence to R.N or A.P.)

R.N.: E-mail: rnovoselov@numerica.us, Telephone: (970) 419 8343

A.P.: E-mail: abpoore@numerica.us, Telephone: (970) 419 8343

this interval, a tracker will have access to the local measurements contained in its soft-level frames, but no mechanism for sharing this information on the network, which necessarily leads to a degradation in SIAP. It must be noted that single frame trackers incur no such latency.

In this paper, we propose a modified version of the Network MFA on Local Data, which mimics the single frame behavior by broadcasting the soft measurement-to-track assignments as soon as they are made. This removes the latency and leaves the remote platforms unaffected by the time it takes to propagate a frame to the back of the moving window. However, since these assignments are “soft”, they are liable to change. In fact, the name “Oops” architecture has been associated with this process because a revocation (“Oops”) message is posted to the network whenever a hard-assignment decision differs from a soft decision. We demonstrate, however, that these revocations are rare in most scenarios. Thus, the “Oops” architecture provides very significant improvements in SIAP with minimal change to the communication load.

This paper is organized as follows. Section 2 reviews the main components of the MFA tracking system. The previously developed MFA network architectures are outlined in Section 3. Section 4 describes the “Oops” architecture and gives an example of its processing sequence. In Section 5, we present experimental results that quantify the performance of this architecture in both air and missile defense simulations. Finally, conclusions are provided in Section 6.

2. MULTIPLE FRAME ASSIGNMENT (MFA) TRACKING SYSTEM

Realistic missile defense problems rarely involve tracking a single target. Therefore, central to the tracking performance is the data association problem of correlating measurements-to-measurements, measurements-to-tracks, or track-to-tracks, such that associated data most likely corresponds to the same physical target. Multiple target tracking methods divide into two broad classes based on their approach to data association, namely single frame and multiple frame methods. The most successful of the multiple frame methods are Multiple Hypotheses Tracking (MHT)² and Multiple Frame Assignment (MFA).^{5,6} The performance advantage of the multiple frame methods over the single frame methods follows from the ability to hold difficult decisions in abeyance until more information is available or, equivalently, the opportunity to change past decisions to improve current decisions. As a result, multiple frame algorithms are better able to handle adverse tracking scenarios caused by false signals, clutter, radar multi-path, residual sensor registration biases, counter-measures, and unresolved closely spaced objects. What is more, data corruption at the local sensor level can be moderated in a composite tracker that uses multiple frame data association.

Numerica has developed a state-of-the-art MFA tracker, which maintains all hypotheses in a window of frames, and formulates the association problem as a multi-dimensional assignment problem subject to certain constraints. The assignment problem is solved through Lagrangian relaxation,⁷ which produces optimal data associations over several data frames. The tracker uses a dual-pane sliding window to maintain track hypotheses for both track initiation and track extension. We denote the size of the window by M/N with $M > N$, where M is the number of frames used for track initiation and N is the number of frames in the sliding window. All M frames within the window participate in track initiation but only the most recent N frames participate in track extension (Figure 1 illustrates a 5/2 window). As the window moves, only the oldest frames of data are permanently assigned to tracks or false alarms. We refer to these permanent tracks as *established* or *firm tracks*. The remaining assignments are *soft* and may be changed based on future frames of data.

The integral part of Numerica’s moving window is the *track tree* which processes all data, stores and maintains all track hypotheses, and moves information through the tracking system. The track tree is an efficient data structure used to store feasible measurements and access track information in the moving window. It allows fast insertion and removal of hypotheses, since sequences are built from subsequences when a new frame is added. The track tree gives efficient access to information since a sequence may be found quickly by starting at the top of the tree and traversing downward. This implies that only a minimal number of sequences need to be compared. The resulting search is faster than a linear search. Thus, when the assignment solver makes a hard decision for a track, incompatible sequences can be quickly found and removed from the hypotheses tree.

Another important component of the MFA tracking system is the frame builder that partitions incoming measurements into maximal proper frames of data in which any particular target is seen at most once. This requirement of being “seen once” is critical to the MFA algorithm and is due to its “unique” assignment approach. For state estimation, the tracker relies on an Interacting Multiple Model (IMM) filter that combines multiple motion models according to a Markov model for the transition between target maneuver states. The tracker also performs various data processing tasks and provides a flexible input/output infrastructure. This allows the MFA algorithm to be easily integrated into a surveillance network via one of the architectures described in the next section.

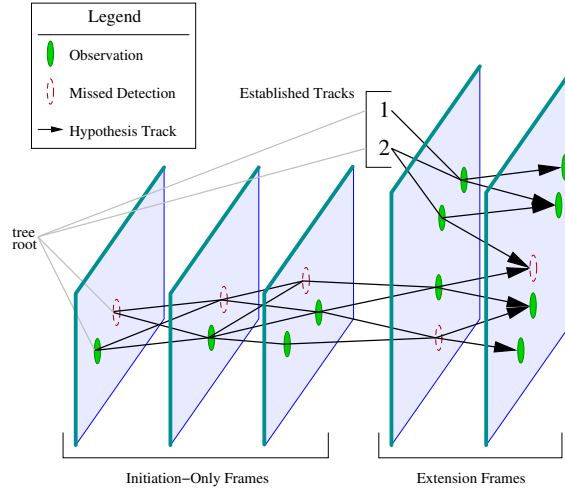


Figure 1. Numerica's double pane moving window. The example shows a 5/2 window.

3. PREVIOUSLY DEVELOPED MFA ARCHITECTURES: BACKGROUND AND TECHNICAL CHALLENGES

In the absence of system latency (e.g., communications and processing) a centralized architecture is probably optimal in that it is capable of producing the best track quality (e.g., purity and accuracy) and a consistent air picture. As such, this architecture is essential as a baseline, even though its exposure to single point failure makes it unacceptable in real systems. Numerica's development of network architectures in support of the multiple frame tracking, therefore, evolved along the following path: (1) Centralized MFA → (2) Replicated Centralized MFA → (3) Network MFA on Local Data and Network Tracks → (4) Network MFA on All Data and Network Tracks → (5) Soft-Level Network MFA ("Oops") architecture. Based on the data collected to date, the soft-level approach appears to provide the best compromise in terms of track accuracy, SIAP, robustness to node failures, and communication load. The implementation of this architecture is discussed in detail in Section 4. First, however, we briefly describe the earlier network MFA architectures, as they form the foundation for the "Oops" algorithm.

3.1. Centralized MFA

The basic centralized tracking architecture is explained in many references such as the book by Blackman and Popoli,² the lecture notes by Drummond,^{8,9} and the article by Moore and Blair.¹ It is reviewed here for completeness. A centralized architecture (Figure 2) is one in which each platform sends its sensor measurements to a central processing unit where the tracking is performed. The tracker is called a composite (i.e., global or network) tracker. Thus, as measurements are received at the central processing unit, they are placed into frames of data and inserted into the moving window.

This architecture is conceptually the simplest and will ideally produce the most accurate data association decisions and tracks. Furthermore, the global tracks are broadcast back to all the platforms, which (in the absence of significant communication delays) guarantees SIAP. The centralized architecture provides a baseline against which other architectures can be measured. However, this architecture suffers from several deficiencies. If the centralized tracker or any communication link becomes inoperable, then the remote platform will lose the capability to track or the quality of local tracks can be significantly degraded. Furthermore, broadcasting track states (state vector and covariance matrix) at each update results in a heavy communication load on the network.

A less well appreciated difficulty with a centralized tracker in a multi-sensor environment is the problem of track initiation. Generally, one requires M/N observations in a moving window to initiate a track; however, if a particular target is not in the field of view of M out of N different sensors, then the target may not be initiated. This problem is moderated by initiating on local data only, as in the distributed architectures below.

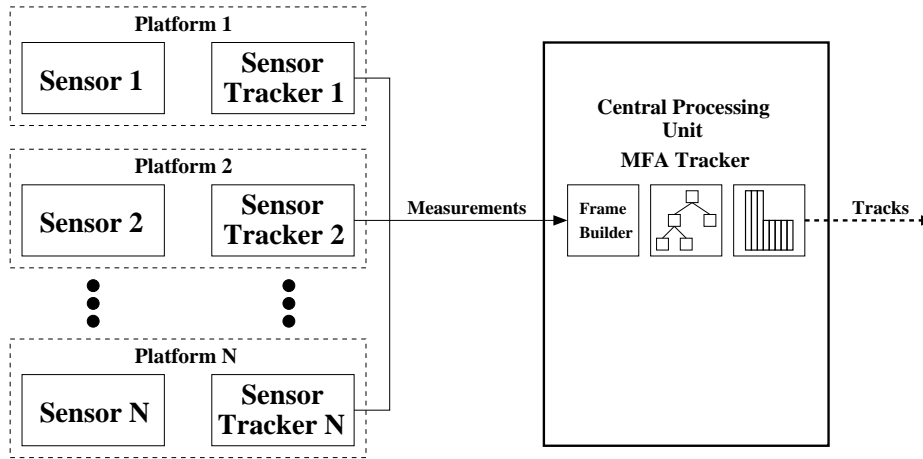


Figure 2. Centralized MFA architecture

3.2. Replicated Centralized MFA

Perhaps the simplest decentralized architecture is the one where a “centralized” composite tracker is placed on each platform. The raw observations generated by the local sensors are fed directly into the platform’s composite tracker, which broadcasts them to the other composite trackers in the network. The communication network used in this scheme is assumed to be fully connected (i.e., there is a direct path between any two nodes). Thus, the composite tracker on every platform has access to the measurements from all local sensors, which it uses for track initiation and maintenance. It does not, however, transmit its track states back to the other platforms.

The Replicated Centralized MFA architecture is illustrated in Figure 3. Its main advantage is that it removes the problem of single point failure. If one or more composite trackers fail to perform correctly, or certain communication links are cut down, the composite trackers on other platforms are still capable of the tracking. Under such circumstances, the composite tracker may not have access to all remote measurement reports, which will cause the tracking performance to degrade gracefully.

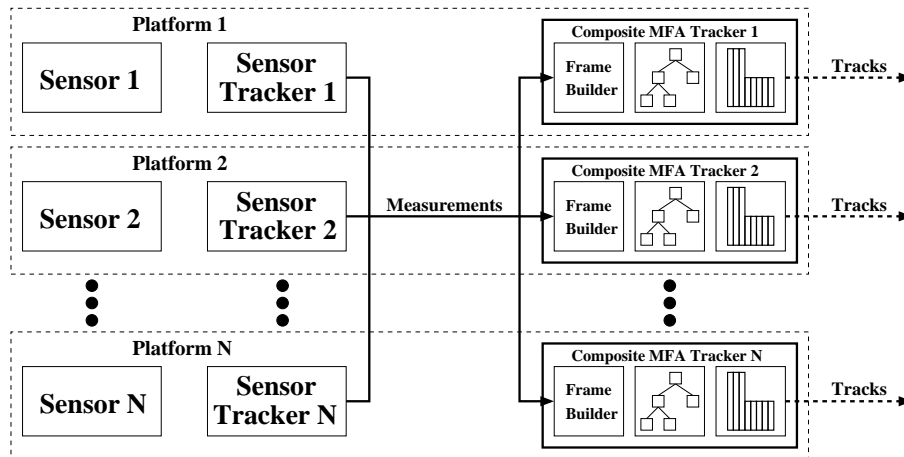


Figure 3. Replicated Centralized MFA architecture

However, the order of observations arriving at different platforms varies due to network latency and band limitations. Each composite tracker is making its own tracking decisions based on the data it receives, regardless of the decisions of other platforms. Moreover, the replicated centralized architecture provides no mechanism for coordinating track numbering between platforms. Thus, in challenging scenarios, SIAP is unlikely to be maintained across the network.

3.3. Network MFA on Local Data and Network Tracks

The Network MFA architecture on Local Data is a variation on the scheme suggested by Kirubarajan, Wang, Bar-Shalom and Pattipati.¹⁰ The composite tracker on each platform only has access to its local data. As a result, the double pane sliding window will contain local frames only, which avoids the initiation difficulties of the centralized / replicated centralized architectures. On each platform, there may be multiple sensors, some of which may have their own sensor trackers. However, it is assumed that raw measurements are available from all sensors, even those with sensor trackers. Figure 4 gives an illustration of this architecture.

Data association decisions are based on the scores of network tracks extending to the local frames. When a measurement becomes firmly assigned to a track (at the back pane of the moving window), the decision is broadcast to all network platforms in the form of Associated Measurement Reports (AMRs). The term AMR is used to denote a message containing a measurement and a composite track ID to which it has been assigned. These messages are used to update the track database on the remote platform.

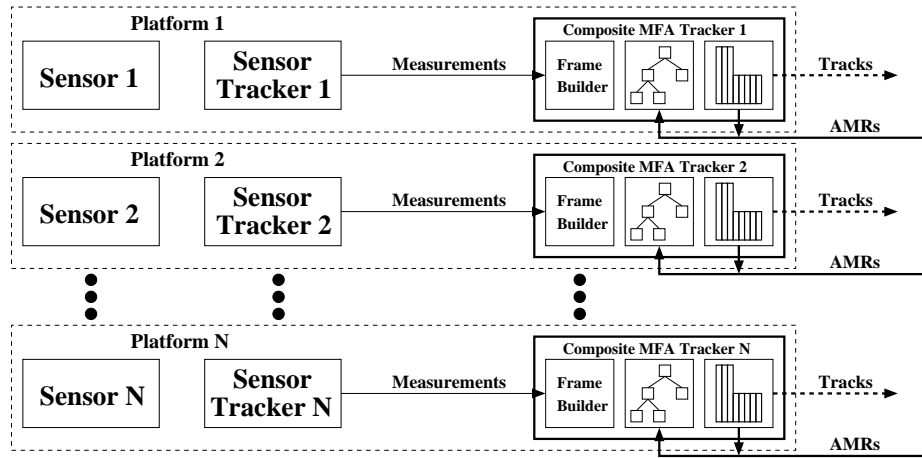


Figure 4. Network MFA on Local Data architecture

The data association in the Network MFA Architecture on Local Data is based on the network tracks, which include both local and remote information. However, the sliding window itself contains only the local frames. This approach is straightforward to implement, but has potential drawbacks in terms of both data association quality and the processing latency (if the local data rate is low). These concerns are addressed in the Network MFA on All Data, discussed in the next section.

3.4. Network MFA on All Data and Network Tracks

The Network MFA on All Data and Network Tracks evolved from the Replicated Centralized architecture by essentially adding the rule that each platform is in charge of assigning its own data to the network tracks. This approach was first suggested by Drummond¹¹ in 1996. There is one composite tracker on every platform. The composite tracker has access to all the data from the sensors on its local platform as well as data from sensors on remote platforms. The double pane sliding window is made up of both remote and local frames of data. The data association decisions are made based on the network tracks and the frames (remote and local) in the window.

Local observations are put into frames by the local composite tracker, which are broadcast to all network platforms. As a result, all composite trackers have the exact same frames. Each composite tracker is allowed to only make firm data association decisions on its own frames (local frames). Thus, the function of the remote frames is to help propagate the local frames through the extension panes of the sliding window until a firm decision can be made. In scenarios with large non-overlapping coverage, this considerably reduces the processing latency and, consequently, improves SIAP. Once the firm decision on the local frame is made, it is reported to the remote platforms in the form of Associated Measurement Maps (AMMs). The AMMs allow the tracker to update its network tracks using the observations in the remote frames.

In real systems, random delays in the transmission network are inevitable. Furthermore, processing time needed to put a remote frame together is different for each tracker. This out-of-order data requires intelligent buffering techniques to prevent processing deadlocks. These are addressed in detail in.¹²

4. SOFT-LEVEL NETWORK MFA (OOPS) ARCHITECTURE

Until recently, Network MFA on Local Data and Network MFA on All Data have been the primary network architectures used by Numerica in both air in missile defense simulations. In realistic networks, they considerably outperform the Replicated Centralized approach in terms of both track accuracy and SIAP. With both of these architectures, trackers make assignment decisions about their local data, and broadcast those associations to the other trackers on the network. Similarly, every tracker is able to accept and receive association messages from any of the remote trackers. The difference between the two architectures is that with “Local Data,” trackers only make their associations using only those frames which are local to the tracker; in the “All Data” architecture, trackers may use remote frames to guide the assignment decisions affecting the local data. In either case, a tracker only makes a firm decision about data which is local, and broadcasts this information to the other trackers.

A limitation of these architectures is that each tracker must wait for a decision to become firm (move to the hard-decision level of the assignment problem) before that decision may be broadcast to peer trackers. This delay causes a degradation in SIAP, since, during that delay, peer trackers have soft information for their local frames, but only hard-level information for the frames from other trackers.

The idea behind the “Oops” architecture is to change this by allowing trackers to broadcast decisions about their local, soft-level associations as soon as they are made. In other words, as soon as the assignment problem completes on a given tracker, the soft-associations are immediately broadcast to all other trackers on the network in the form of Soft Associated Measurement Reports (SAMRs). The use of SAMRs has led to a very significant (as much as 90 %) SIAP improvement in some scenarios, as shown in Section 5.

However, because the information broadcast is “soft,” it is liable to change. To allow for this, each tracker records the soft association decision it makes, and keeps that information so that it may be compared to a hard decision later on. When the comparison is made, if the hard-association turns out to be different from the soft-association, the tracker broadcasts an “Oops” message, telling all other trackers on the network that it has changed its mind. The trackers respond to this message by finding the altered track in their local databases, and rebuilding the track-tree using this new information. “Oops” messages (association revocations) are relatively rare, as they should be, meaning that most of the benefit of the “Oops” architecture comes from the ability to broadcast track associations sooner than they otherwise would have been.

Implementation of the “Oops” architecture requires that a new Track Revocation Monitor object be connected to the sliding window. This object monitors the soft-assignment decisions emitted by the tracking kernel and records them. When a hard-assignment decision which disagrees with a soft decision is noticed, the monitor emits a signal indicating that a “Track Revocation” has occurred, providing the track ID, frame ID, old, and new indices. (A revocation may also correspond to a change to-or-from a missed-detection in that frame.) Using this data, Oops messages may be built and broadcast to the other trackers when necessary.

Figure 5 illustrates the processing sequence in the “Oops” architecture. If communication and processing delays are ignored, then all frames arrive to the different platforms in exactly the same order. We use this ideal case to simplify our discussion. We further assume that all trackers are well synchronized. There are three platforms i , j , and k shown on the graph. A track extension window of size 4 is chosen for all three platforms.

Initially, the window on each platform contains no frames of data. We start with the same set of network tracks on platforms i , j , and k , which are represented by dark black dots in front of the track extension window. Suppose a local frame on platform i is ready to be processed. The frame is inserted into the extension window, and the track-tree is extended. After setting up the assignment problem, platform i makes soft data association decisions on this first frame based on the network tracks. These decisions are broadcast as a set of SAMRs to platforms j and k . The composite tracker on platform i prunes its track-tree and moves the sliding window forward. The composite trackers on both platform j and k update their network tracks based on the SAMRs received and rebuild the track-trees in their extension window.

After all composite trackers finish processing the local frame from platform i , the network tracks are still the same in the sense that they all have the same observation history associated with each track. At this point new frames on platforms

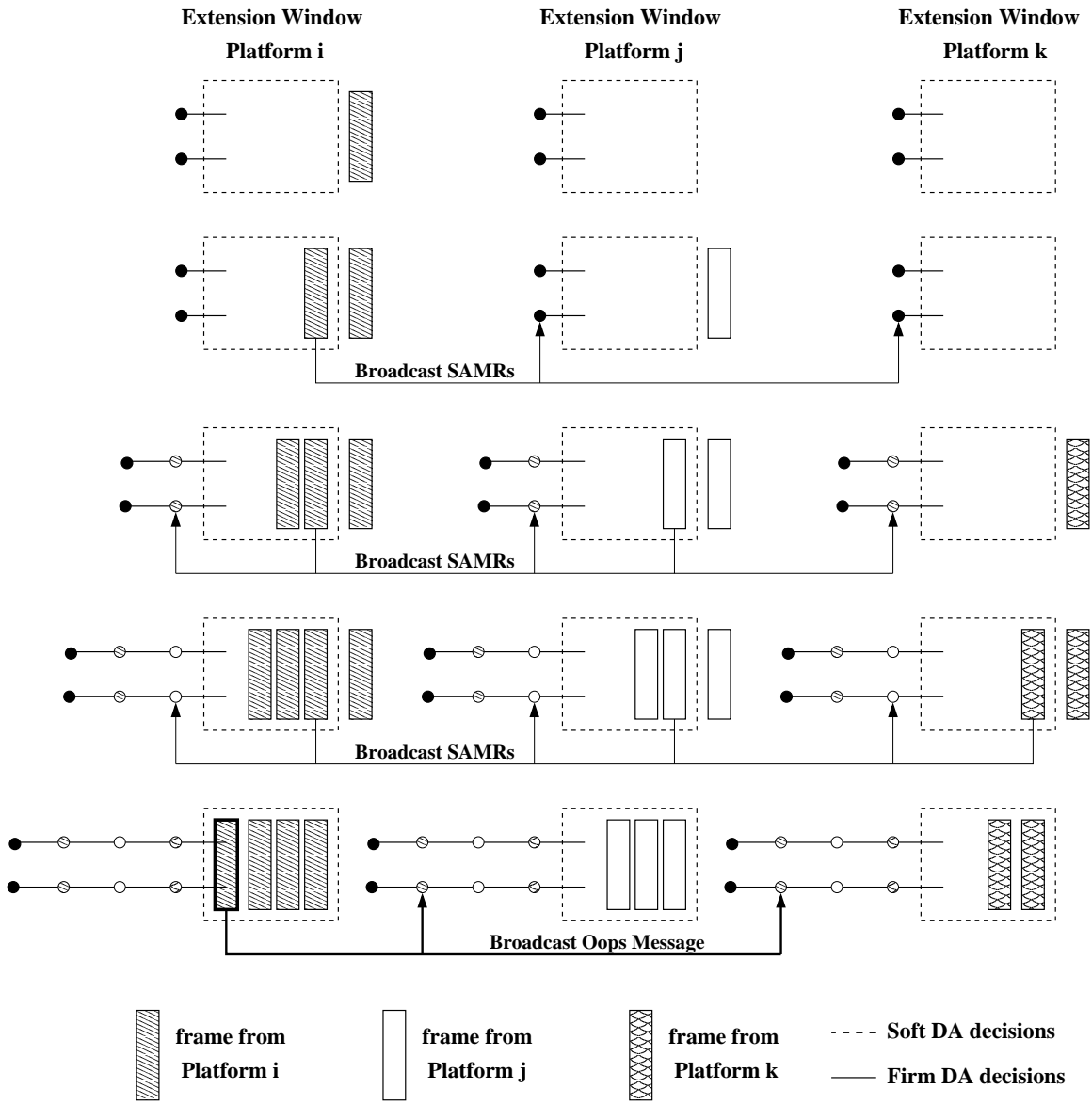


Figure 5. Processing sequence in the "Oops" architecture

i and j are ready to be processed. The composite trackers on those platforms insert the frame into their extension windows, update the track-tree, make the data association decisions, and broadcast them to the network. The moving windows on platforms i and j are moved forward after that. All platforms update their network tracks and the track-trees according to the SAMRs received. Similarly, when a local frame from platform k is ready, the composite tracker on platform k makes its soft data association decisions and broadcasts them to platform i and j .

Finally, another frame on platform i is inserted into the extension window. The assignment problem now includes not only this last frame and the network tracks, but also the very first frame (which is now at the back of the window). When the assignment problem is solved, the SAMRs corresponding to the new frame are once again sent to the other platforms. However, at this point, an irrevocable (hard) data association decision is made on the frame at the front of the window. The Revocation Monitor will compare this decision with the original assignment. If the two match - there's no work to be done. However, in this case we assume that the assignment has changed and the "Oops" message must be sent to platforms j and k . The composite tracks on those platforms use the new information to re-filter the corresponding tracks and to rebuild the track-tree.

It can be concluded from Figure 5 that, in the absence of communication delays, the network tracks are exactly the same across the platforms in the sense that they have the same observation history. Imperfections in the communication link result in the out-of-order messages, which complicate matters significantly. Out-of-order SAMRs require refiltering and rescoreing. In the worst case, the observation contained in an SAMR may fall out of the extension window. With Oops messages, several exceptional cases may also arise in their processing and must be considered:

1. If the track cannot be located, the Oops message is completely ignored.
2. If the old observation contained in the message cannot be found in this history, no observation will be removed, but the tracker will still attempt to insert the new observation.
3. If the new observation causes the track's score to become positive or to increase (get worse) by a significant amount, the new measurement will not be added to the track.

Nevertheless, with adequate exception handling and intelligent filter caching/buffering strategies, the "Oops" architecture achieves both good run-time performance and near-perfect SIAP in realistic network environments.

5. SIMULATION RESULTS

In this section we present simulation results to quantify the performance of the "Oops" architecture. We compare this architecture to its predecessor - Network MFA on All Data, using both track accuracy and SIAP metrics. Since the architectures have been implemented within both the Joint Composite Tracking Network (JCTN) and the Ballistic Missile Defense (BMD) Benchmark, we choose a representative scenario from each simulation environment.

5.1. Air Defense Scenario (JCTN)

The air defense scenario used in this study contains four fighter aircraft and is 1200 seconds in length. There are four ship platforms defined, each with an S-band Phased Array Radar. Figure 6 illustrates the target and platform trajectories and depicts the merging of the fighters. The composite trackers in both architectures apply the MFA algorithm. An IMM filter with two nearly-constant velocity (NCV) models (noise levels 1 and 1000) is used.

Both residual gridlock biases and communication delays are introduced into the scenario. The biases are "pair-wise relative" and constant over the course of a single Monte Carlo run. The communication delays are modeled as having both a deterministic and a stochastic component. The deterministic delays are unique for each pairing of originating platform to destination platform. The stochastic component results from queuing at the input of the originating node, and is a random quantity. Communication buffer triage and imperfect transmission are also modeled, such that approximately 1 out of 10000 messages is dropped.

Figure 7 presents the composite track state estimate differences. This is a SIAP metric which most clearly demonstrates the benefits of broadcasting soft-level data association decisions. This metric measures the Euclidean distances between the position estimates of tracks held by pairs of platforms for composite tracks with the same track number. As shown in the figure, the "Oops" architecture affords a 62.1% improvement over the Network MFA on Local Data. This

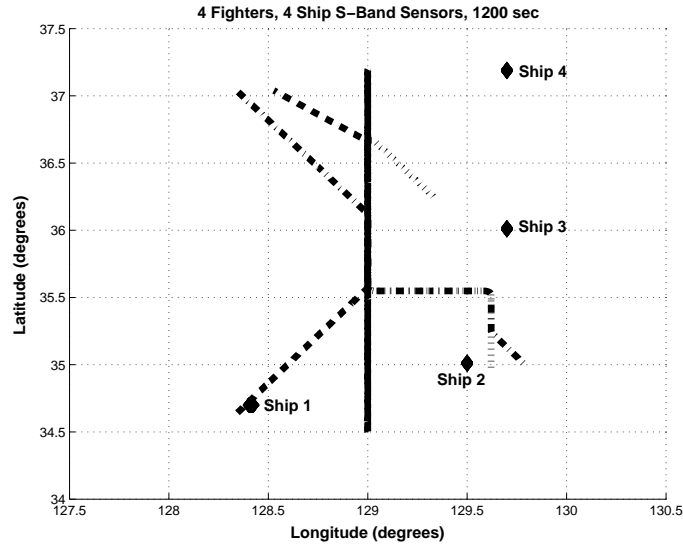


Figure 6. JCTN Scenario Description

number is representative for a contentious scenario, where scores going into the assignment problem are very close. The contention comes from both the imprecise nature of the motion models and the closeness of the aircrafts' trajectories. In such scenarios, the revocation (Oops) messages are not altogether rare. (In this particular scenario, there is approximately 1 revocation for every 18 frames processed). In simpler scenarios where soft-level decisions are unlikely to change, the SIAP improvement is even greater. This is shown in the next section.

Composite track accuracy is illustrated in Figures 8 and 9. It is measured as an RMS error between the reported and the true position/velocity of the target, averaged over all Monte Carlo runs. For the particular platform/target combination shown, the "Oops" architecture provides a 30.1% decrease in position error and a 25.0% decrease in velocity error. Note that it avoids a spike in both position and velocity errors at approximately 120 sec, when fighters 3 and 4 merge trajectories. This improvement in track accuracy is consistent across all platforms and is seen in most scenarios. Mainly it stems from the fact that reduced latency in receiving remote data association decisions keeps the local tracks on each platform considerably more up-to-date. As a result, the track states reported at metrics calls are generally more accurate. Also, the SAMR message arriving earlier than the AMR otherwise would, alters the score of the existing track. This, in turn, affects the assignments made in the local window. In dense scenarios such as this one (and assuming that the SAMR is not subsequently revoked), this also leads to accuracy gains.

Finally, Figure 10 illustrates the composite redundant mean ratio as a function of time, which is an important track continuity metric. This ratio is equal to the number of composite tracks assignable to a truth object divided by the number of valid composite tracks. A ratio of one is the desired value, indicating exactly one track per every truth object. The "Oops" architecture has noticeably fewer redundant tracks (again, due to decreased latency and better data association), although some redundancy present in both architectures. In particular, in many Monte Carlo runs, both architectures generate redundant tracks at 480 seconds when all planes merge, and at 977 seconds when they separate. These maneuvers present a challenge to the IMM filter, which includes only the NCV motion models. Still, the redundant tracks are mostly due to the erroneous tracks produced by the local sensor. Since both the Network MFA on Local Data and the "Oops" architecture use only the local data in multi-frame assignment, they are not always able to moderate this data corruption. The Network MFA on All Data fairs better in that regard. It must also be noted that a spike at approximately 275 seconds represents not an actual redundant track, but rather an instance when two tracks gate exclusively with the same truth object. Both of these tracks are, therefore, assignable, but only one will be selected and considered valid. This is a relatively rare occurrence, but it does show up on the redundancy metric.

5.2. Missile Defense Scenario (BMD Benchmark)

In this simulation, there are four sensor platforms and four missiles: two single-stage missiles (M600s), and two two-stage missiles (M1250s), each of which contain an attitude control module (ACM) for the associated RVs. An MFA composite tracker is situated on every platform and all trackers use identical parameter sets. For state estimation we rely on a variable structure IMM filter, which toggles between boost, midcourse, and re-entry motion models. The midcourse model set includes two continuous gravity compensated NCVs, with noise levels set to 0.0001 and 1.0. As in the previous scenario, residual biases and communication delays/triage are enabled. This scenario forces the local trackers to look in geometrically non-overlapping locations of sky to collect measurements on targets. Moreover, sensors are generally unable to cover the entire trajectory of every missile, meaning that composite trackers sometimes rely exclusively on the remote information to maintain a complete air picture. In such instances, the latency in receiving remote measurement-to-track association can be especially detrimental to tracking performance.

Figures 11 and 12 present, respectively, the position and velocity components of the track state estimate difference. The comparison is made between two X-band sensors. The data association problems are simpler here than in the air defense scenario owing to the ballistic nature of the motion, although the closely spaced objects (CSOs) complicated matters somewhat. In any case, the assignment revocations are extremely rare (approximately 1 “Oops” message sent for every 52 assignments), which leads to a very significant improvement in SIAP. For the platform pair shown in the figures, the position estimate difference is reduced by 90.3% through the used of soft-level AMRs. This figure becomes 84.3%, when averaged among all possible platform pairings.

Figures 13 and 14 demonstrate that in a realistic missile defense simulation the “Oops” architecture provides not only better SIAP, but also superior track accuracy when compared to the Network MFA on Local Data. Most notably, it avoids the temporary degradation in accuracy around 670 seconds, which corresponds to the start of re-entry. Due to the incomplete coverage, the re-entry phase of the mid-range missiles is generally observed by only one or two sensors. The remaining platforms wait for AMRs to signal a phase shift. In the meantime, when propagating the tracks to the time of the metrics calls, these platforms use the outdated midcourse motion models, resulting in very inaccurate track states. The “Oops” architecture shortens the delay in receiving the AMRs by supplying soft-level data association decisions. In doing so, it significantly improves the accuracy of reported tracks during re-entry. Note that the position and velocity error profiles shown on the figures are for one of the M1250 reentry vehicles (RVs), which represent a more challenging tracking problem. The tracks on the M600 missiles are considerably more accurate.

The redundant track ratio is shown on Figure 15 . The object reportability criteria used in the BMD Benchmark must be taken into account in order to correctly interpret this data. Since this scenario does not simulate debris from the missiles, the total number of targets for which the tracker can receive measurements is twenty-two (22). However, not all detected targets are scorable (a.k.a. reportable). In fact, only 17 objects are used by the metrics package for scoring measures of performance (MOPs). Thus, many of the redundant tracks seen on Figure 15 are due to measurements received by the local sensor on non-reportable objects. The actual redundant tracks are also present in both architectures, but the “Oops” architecture has significantly fewer of them. This is due to the improvements in track accuracy and data association quality resulting from the use of soft-level AMRs.

6. CONCLUSION

In this paper, we introduced the latest in a series of decentralized network architectures developed in support of the multiple frame tracking. This architectures affords very significant improvements in the consistency of the air picture by broadcasting soft data association decisions as soon as they are made. Trackers have the ability to revoke these decisions (via an “Oops” message) if they differ from the firm associations made at the back of the moving window. We have shown, however, that such revocation are extremely rare even in challenging scenarios. This work presented two simulations studies, which compared the “Oops” architectures with its predecessor - the Network MFA on Local Data. In both studies, the new algorithm emerged as a clear winner in terms of track accuracy, track consistency, and, especially, in its ability to maintain SIAP.

ACKNOWLEDGMENTS

This work was supported in part by the ONRL Contract Number N00014-01-C-0212, the AFRL Contract Number F33615-03-C-1434 (SBIR Phase II), and the AFOSR Contract Number FA9550-04-1-0222.

REFERENCES

1. J. R. Moore and W. D. Blair, "Practical aspects of multisensor tracking," in: *Multitarget/Multisensor Tracking: Applications and Advances III*, 2000.
2. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Norwood, MA, 1999.
3. S. Lu and A. B. Poore, "Network-centric mfa architectures," in *Cooperative Control: Models, Applications and Algorithms*, P. Pardalos, R. Murphy, and S. Butenko, eds., pp. 187–213, Kluwer, 2002.
4. A. B. Poore, S. Lu, and B. Suchomel, "Network MFA tracking architectures," in *Proceedings of SPIE Conference on Small Targets 2001*, Oliver E. Drummond Editor, 2001.
5. A. B. Poore, "Multidimensional assignment formulation of data association problems arising from multitarget tracking and multisensor data fusion," *Computational Optimization and Applications* **3**, pp. 27–57, 1994.
6. A. B. Poore and A. J. R. III, "A new Lagrangian relaxation based algorithm for a class of multidimensional assignment problems," *Computational Optimization and Applications* **8**, pp. 129–150, 1997.
7. A. B. Poore, Jr., "Method and system for tracking multiple regional objects by multi-dimensional relaxation." US Patent Number 5,537,119, issued 16 July 1996. (Assignee: Colorado State University Research Foundation, Fort Collins, CO).
8. O. E. Drummond, "A hybrid fusion algorithm architecture and tracklets," *Proceedings of SPIE* **3136**, pp. 485–502, 1997.
9. O. E. Drummond, "Multiple sensor, multiple target tracking lecture notes," 1999.
10. T. Kirubarajan, H. Wang, Y. Bar-Shalom, and K. R. Pattipati, "Efficient multisensor fusion using multidimensional assignment for multitarget tracking," *SPIE Proceedings* **3374**, pp. 14–25, 1998.
11. O. E. Drummond, "Private communication," 1996. published in a generally unavailable reference.
12. S. Lu, *Network Multiple Frame Assignment Architectures*. PhD thesis, Colorado State University, 2001.

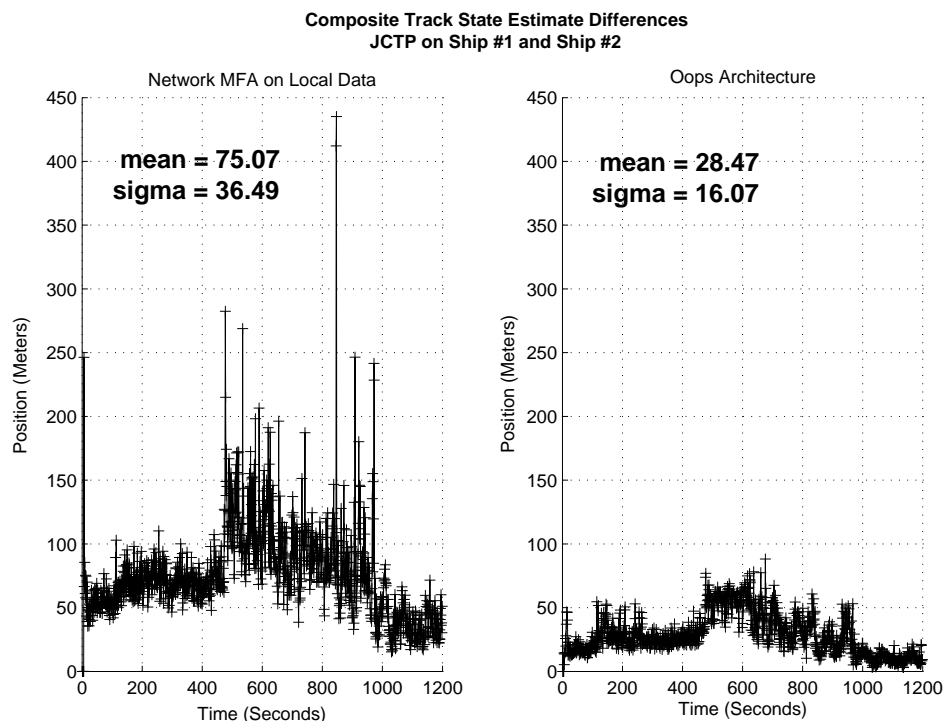


Figure 7. JCTN scenario: composite track state estimate differences

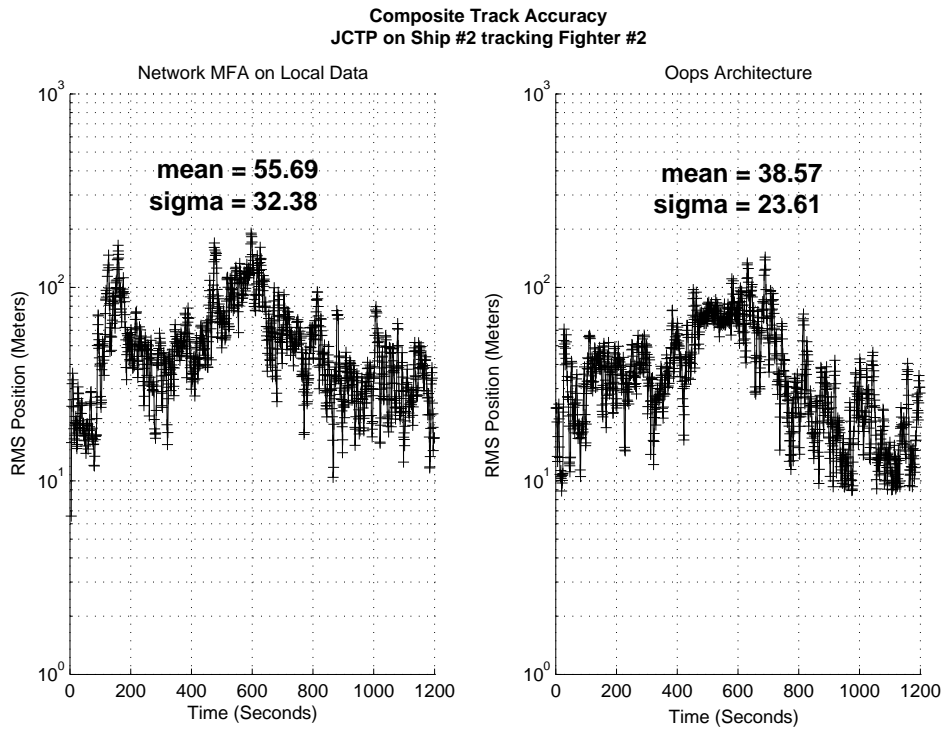


Figure 8. JCTN scenario: RMS position error metric

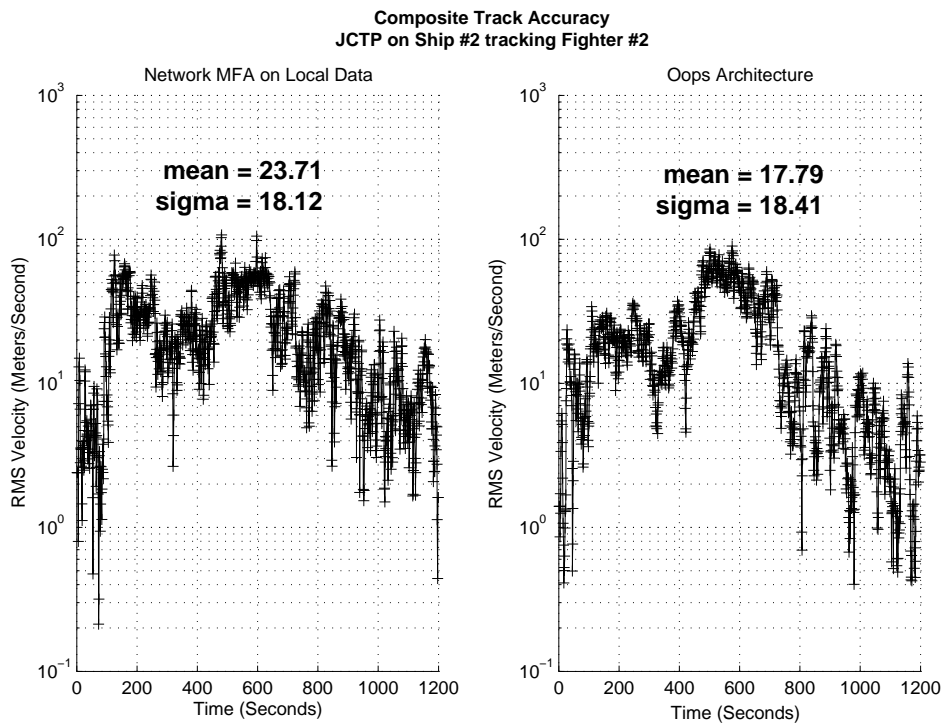


Figure 9. JCTN scenario: RMS velocity error metric

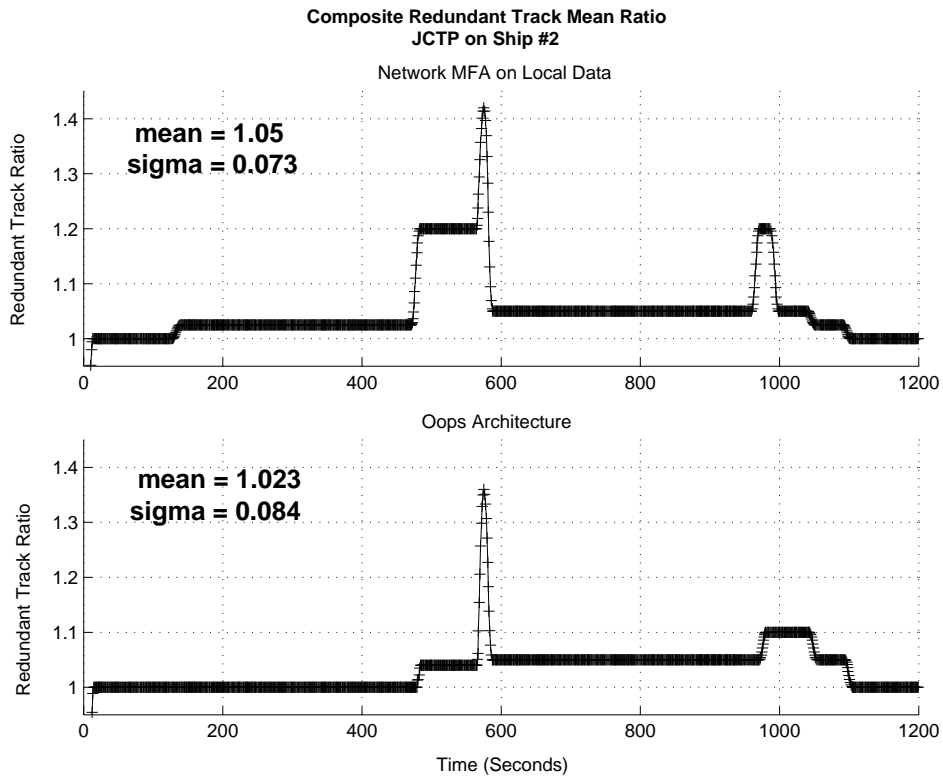


Figure 10. JCTN scenario: composite redundant track ratio metric

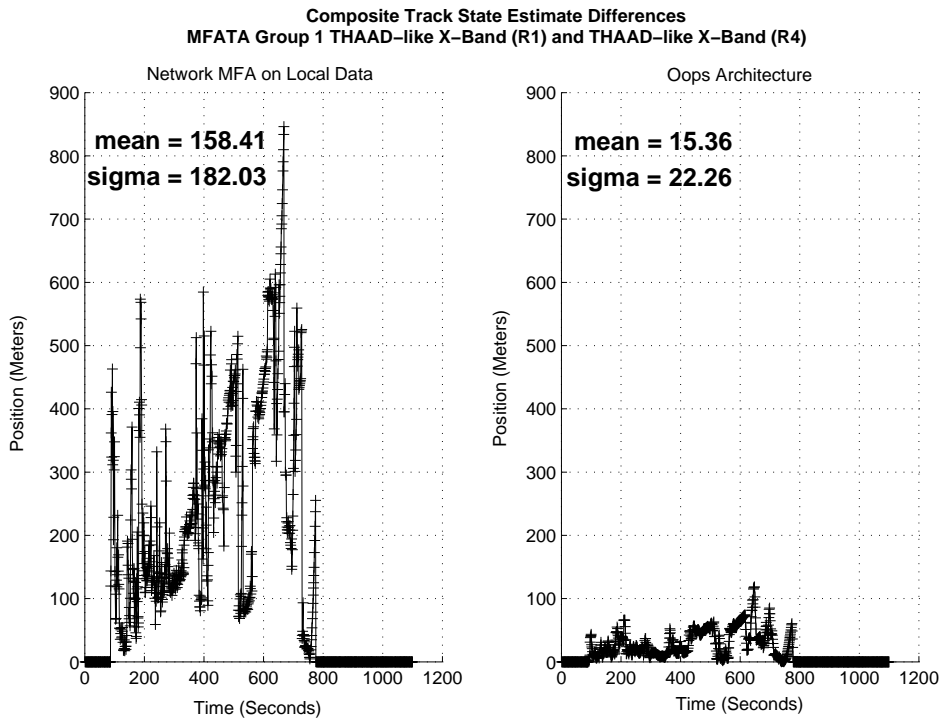
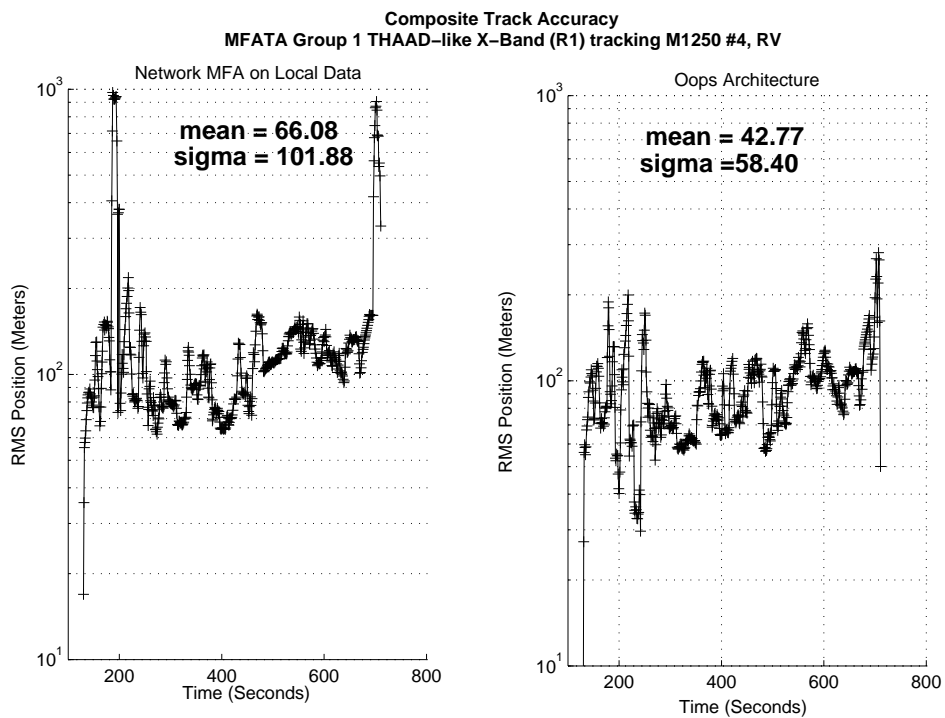
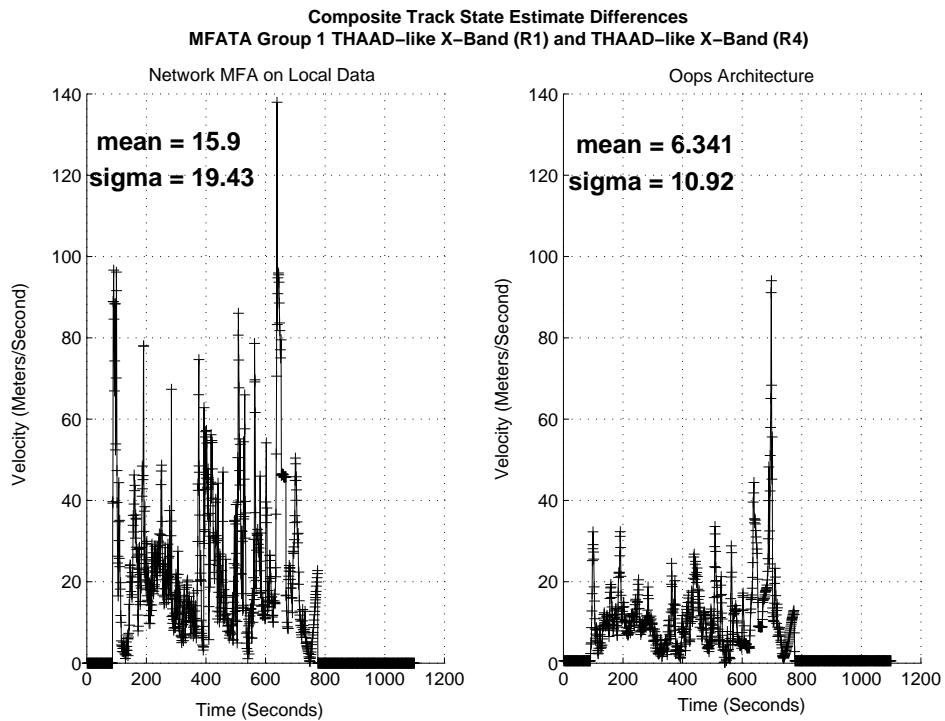


Figure 11. BMD scenario: composite track position estimate differences



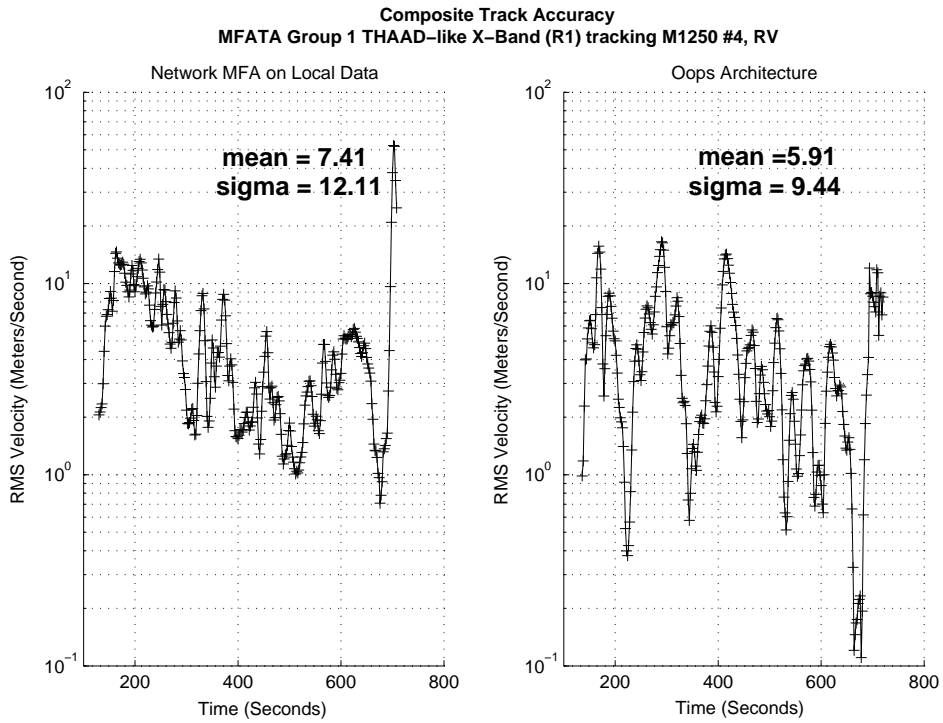


Figure 14. BMD scenario: RMS velocity error metric

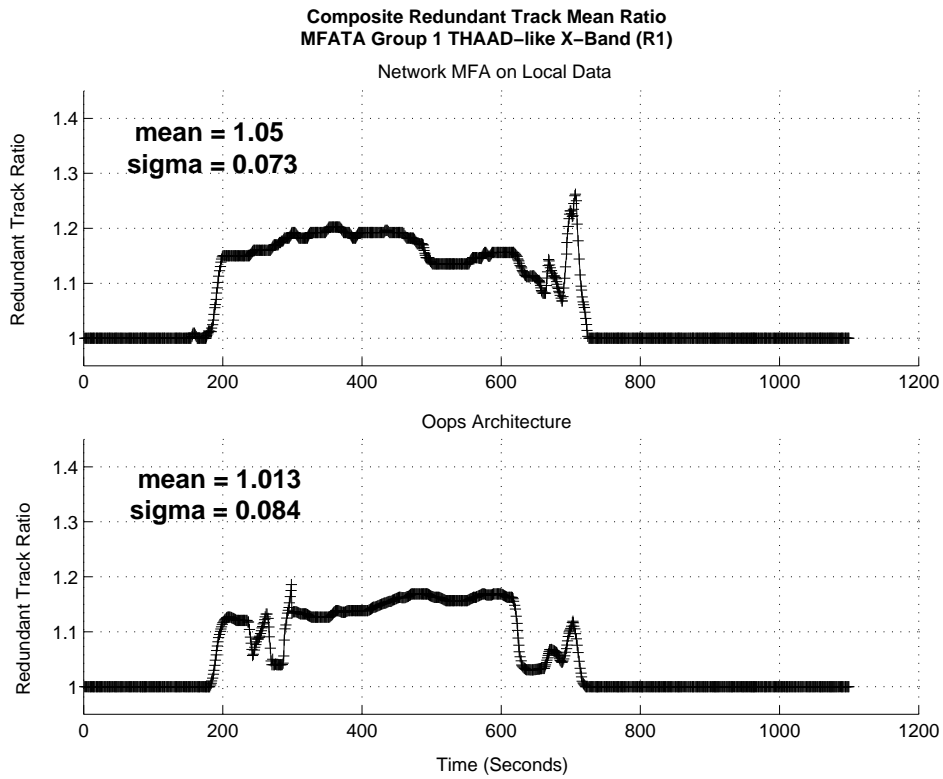


Figure 15. BMD scenario: composite redundant track ratio metric