5/30/95

SANDIA REPORT

SAND95-0798 • UC-705

Unlimited Release

Printed May 1995

Measures of Effectiveness of BMD Mid-Course Tracking on MIMD Massively Parallel Computers

J. P. VanDyke, J. L. Tomkins, M. D. Furnish

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SAND95-0798
Unlimited Release
Printed May, 1995

Distribution
Category UC-705

Measures of Effectiveness for BMD Mid-Course Tracking on MIMD Massively Parallel Computers

J. P. VanDyke and J. L. Tomkins
Parallel Computing Science Department

M. D. Furnish
Experimental Impact Physics Department

Sandia National Laboratories

Albuquerque NM 87185

Abstract

The TRC code, a mid-course tracking code for ballistic missiles, has previously been implemented on a 1024-processor MIMD (Multiple Instruction - Multiple Data) massively parallel computer. Measures of Effectiveness (MOE) for this algorithm have been developed for this computing environment. The MOE code is run in parallel with the TRC code. Particularly useful MOEs include the number of missed objects (real objects for which the TRC algorithm did not construct a track); of ghost tracks (tracks not corresponding to a real object); of redundant tracks (multiple tracks corresponding to a single real object); and of unresolved objects (multiple objects corresponding to a single track). All of these are expressed as a function of time, and tend to maximize during the time in which real objects are spawned (multiple reentry vehicles per post-boost vehicle). As well, it is possible to measure the track-truth separation as a function of time. A set of calculations is presented illustrating these MOEs as a function of time for a case with 99 post-boost vehicles, each of which spawns 9 reentry vehicles.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED *JR*

MASTER

Table of Contents

1.0	Introduction	7
2.0	Main Elements of the tracker/correlator calculation	8
3.0	MOEs and the TRC	9
3.1	Descriptions of Measures of Effectiveness	9
3.2	The TRC Simulation Package	11
4.0	Ramifications of the MPP Computing Environment	12
5.0	Sample Problems	13
6.0	Conclusions	21
	References	22

Figures

1	Ghosting due to erroneous range assignment.	9
2	Schematic of MOE study	10
3	TRC parallel implementation	12
4	“Double Node” implementation of MOE analysis	14
5	Measures of Effectiveness for actual sensor noise of 10 μ rad. Assumed sensor noise varied from 7 to 50 μ rad.	16
6	Measures of Effectiveness for actual sensor noise of 10 μ rad. Assumed sensor noise varied from 7 to 50 μ rad.	17
7	Measures of Effectiveness for actual sensor noise of 50 μ rad. Assumed sensor noise varied from 25 to 250 μ rad.	18
8	Time-averaged values of Measures of Effectiveness	19

Tables

1	Assumed and Actual Sensor Noise Parameters Used	15
2	Time averaged values of Measures of Effectiveness for TRC with 99-9 threat	15

This page intentionally left blank

Definitions

Cluster: A group of tracks, maintained and updated by a single node (processor). A node may maintain more than one cluster.

Constructed Track Complex: A "best guess" by the tracker-correlator code for the trajectory of an object. Usually a composite of several tracks from one or more types of sensors.

Gate: A program which assigns sensor reports to clusters

Hypothesis: A possible object trajectory, inferred from a limited number of sensor reports

MHT: Multiple-Hypothesis Tracking; a methodology whereby, within a cluster, one sensor report can be assigned to multiple tracks and multiple sensor reports can be assigned to one track. Tracks are then pruned as they are found to be clearly erroneous.

MIMD: Multiple Instruction - Multiple Data (computer type capable of parallel processing)

MOE: Measure-of-Effectiveness, i.e. a metric of success or accuracy

Node: A processor in an MIMD machine

Object: An item to be tracked (here a post-boost vehicle, a reentry vehicle or a decoy)

Observation: Sensor report

Sensor: A device for measuring position of objects. In the present study, assumed to be mounted on a satellite and capable of measuring azimuth and elevation to a prescribed precision, but incapable of range measurement.

Track: Calculated trajectory of an object described by position and velocity at a point in time and a covariance matrix.

TRC: A tracker/correlator code originally developed by D. H. Wagner & Associates and Ball Systems Engineering for the Naval Research Laboratory (serial version), then modified for MIMD processing at Sandia National Laboratories.

Truth: The trajectory of an object, used to generate sensor reports to drive the TRC algorithm.

Measures of Effectiveness for BMD Mid-Course Tracking on MIMD Massively Parallel Computers

1. Introduction

A system for tracking ballistic missiles relies on the successful integration of heterogeneous sensor data into closely defined paths identifiable with real objects. The computational component of such a system, known as a tracker/correlator, receives sensor data and attempts to construct model paths consistent with that data and with physics constraints. The available sensor data includes time-correlated observations of elevation, azimuth and (for certain detectors) range to an object, and is interpreted in light of sensor positions. The process of combining observations (sensor reports, or position and velocity information at known times) into tracks (inferred orbital paths of objects) is referred to as tracking. As further sensor reports become available, the tracks must be continuously updated and subjected to consistency tests.

Measures of effectiveness (MOEs) provide a metric of the performance of tracker/correlator systems, and must be established to assess the suitability of such systems for use in real situations where validation will not be possible. A set of MOEs for such a system are as follows:

- How many objects are missed?
- How many ghost tracks are produced (tracks not corresponding to an object)?
- How many objects are unresolved (multiple objects corresponding to a single track)?
- How many tracks are redundant (corresponding to same object)?
- How closely are objects tracked?
- How “pure” are the tracks (did sensor reports used to generate a path all come from the same object)?

In general, such MOEs are applied to the algorithm by supplying to the algorithm mock sensor inputs generated from “truths,” allowing the algorithm to calculate models of truths, then comparing the results with the original truths. Note that these MOEs are only measured in a simulation, where we can perform this comparison.

The present study focuses on tracking effectiveness during the mid-course phase of a missile flight, which is the most critical period for establishing the missile course and probable target and hence the most effective response. The identifiable flight path during the

launch phase does not adequately define the target, while insufficient time exists during the terminal phase to assess the threat, determine the most suitable response and effect that response. For purposes of the present discussion, we are concerned with scenarios of 1,000 - 2,000 objects to be tracked.

The tracker/correlator selected for this study is a modified form of the Naval Research Laboratory computer code, TRC. This algorithm was available and provided a convenient, computationally representative platform to use in performing MOEs.

The TRC was modified at Sandia for use on multiprocessor platforms, a tractable process because the calculations are structurally appropriate for a parallel environment (Multiple Instruction Multiple Data, or MIMD). Computational speed is the major advantage of this implementation. The speedup may range up to 300 - 400 (30-40% utilization on 1024 processors). These calculations may also be performed on conventional workstations, and the speeds available on such platforms now approaches what is required for present scenarios. Specifically, the track matching and projection for a real application must be done with speed sufficient to allow several iterations of the tracking and projection during the flight of the object in question, followed by time for a counteraction decision to be made and implemented. In other words, it must be performed faster than real time. For example, one might envision a 20 minute total action time, of which 2 - 3 minutes is dedicated to realizing that a threat exists, 8 - 10 to performing tracking and projection calculations, 1 -2 to making response decisions, and 5 to implement those decisions to conclusion. The actual schedule is likely to be tighter than the above.

2. Main elements of the tracker/correlator calculation

The basic function of a tracker/correlator algorithm is to convert sensor reports into models of trajectories of objects, *i.e.* tracks. Sensor types may be grouped as active (generally, ground-based radar), which contains range information, and passive (generally, space-based infrared), which does not. Sensor reports have uncertainties due to noise and other factors; these uncertainties propagate to the accuracy of the tracks. These reports also must be interpreted in light of the position of the sensors, which may be time-dependent as for orbiting sensor systems.

A major issue is associating sensor reports with previously established tracks. Similarly for track fusion, matching tracks from different sensor systems is subject to analogous association errors. In certain scenarios, several thousand real objects may be involved. Two types of association errors may occur: redundant or ghost tracks and missed objects.

A ghost is a well-defined track without a corresponding object. A redundant track is a second track associated with an object. Passive sensors are particularly susceptible to ghosting when objects pass close to their focal plane [Roecker, 1992]. For example, erroneous range assignments may be made to a continuing track after the corresponding object has made a skew passage with another object, as shown in Fig. 1.

Figure 1. Ghosting due to erroneous range assignment.

A missed object is an object without a track. It may correspond to a failure of a sensor to pass on a signal recognizable as an object to the computer. It may also correspond to an improper association of sensor reports. In a simulation, these can be detected by a comparison of tracks and truths, while in a real situation such information is not available.

The TRC tracker/correlator algorithm is an example of the Multiple (competing) Hypotheses Tracking (MHT) Algorithm type; it keeps competing hypotheses open until further data are available. The MHT character of the TRC can lead to very rapid growth in the number of tracks being maintained and propagated. To minimize the number of tracks retained, the TRC prunes tracks in a number of ways. Tracks that are not updated within a specified time period are pruned. Also, within a cluster (set of tracks in close proximity; see Section 4), tracks are compared to each other and duplicates are pruned. Typically, the MHT process used in the TRC results in three to five times as many hypotheses being retained as there are real objects. Here, the term “real objects” refers to constructed track complexes, which are essentially best guesses in the TRC for the actual objects being tracked. These outnumber the actual objects by approximately 20%. For dense scenarios, this ratio of hypotheses to real objects can be much larger, while for sensors with high resolution the ratio is smaller. Another MHT algorithm, produced by ALPHATECH [Allen, 1992], uses 3 (not necessarily consecutive) misses as a criterion for discarding hypotheses; maintaining far fewer tracks. The danger in applying an excessively stringent criterion for track retention is that actual object tracks may be excluded.

3. MOEs and the TRC

3.1 Description of Measures-of-Effectiveness

A Measure of Effectiveness (MOE) study of a tracker/correlator algorithm proceeds as outlined in Figure 2. A generator produces “Truths,” or descriptions of trajectories of objects (“targets”) to be tracked. These ballistic trajectories are calculated using such corrections as the J_2 term [VanDyke and Tomkins, 1992]. From these Truths, sensor report simulations are generated. Noise and imprecisions are introduced. The tracker/correlator algorithm to be tested is then given these sensor reports in a sequential manner. It constructs competing hypotheses of object orbits, choosing amongst these its “best guesses” for the actual trajectories, that data which would be passed on to situation commanders in an actual situation. This choosing is an ongoing process, and choices may be changed as further sensor reports become available. The tracker/correlator assumes the same ballistic physics as does the track generator, as it must if the tracker/correlator output vs. Truth correspondence is to reflect the algorithm success [VanDyke and Tomkins, 1992]. Finally, the

Figure 2. Schematic of MOE study

MOE code compares the tracks generated by the tracker/correlator with the Truths. MOE data such as the following is available with appropriate run parameters:

- How many objects are missed at a given time?
- How many tracks are duplicates or ghost tracks at a given time? When more than one sensor report is assigned to an existing track, one or more new hypotheses are initiated; this is necessary to allow for spawning of objects, but can cause duplicate tracks.
- How many objects are unresolved at a given time? Generally an issue for closely spaced objects, these may be due to failure of the sensors to discriminate multiple closely-spaced objects (sensor noise).
- How “pure” are the tracks? If tracks are inconsistently assigned to objects, their predictive value is diminished. This issue is especially important in the MIMD context where erroneous gating decisions may result in clusters artificially gaining or losing objects.
- How closely are objects tracked? This may be a statistical cumulation of track/truth separation over the trajectory, or a measure of deviation at some time, such as the end of the mid-course phase.

We note that there are other measures of effectiveness which are not calculated here, such as state estimation bias and error and filter covariance credibility. As well, non-tracking considerations such as object classification and discrimination, signal processing and system performance are not discussed in the present paper.

The MOE code is the subject of the present section. This code has two main tasks. The first is to determine which track to associate with which truth. In a multi-target tracking (MTT) problem with large numbers of targets, this can be a very difficult task. The second task is to calculate the distance between track and truth.

The main ways tracks can be associated with truths are:

1. Associate the nearest truth for each track. This is the primary association method used in the present study, and is easy in a distributed data structure such as those used in massively parallel computing environments (see next section).
2. Associate the nearest track for each truth. This is also used in the present study, although it is more difficult and inefficient in a distributed data structure.
3. Do a "global optimal assessment," i.e. find the association matrix that minimizes the overall error [Drummond and Fridling, 1992]. There are many possible variations of this option.

The task is made more difficult by a real-life "fuzziness," which is modeled here. "Missed objects" and "unresolved objects" both describe true objects without uniquely associated tracks, and these situations occur. There may be objects a significant distance from any tracks, or there may be fewer tracks than objects for the case of a relatively tightly clustered group of objects. These represent end-member cases; intermediate situations may also be found in real tracking contexts.

Specific real-life factors which may be introduced include physical parameters and tracking calculation parameters. Examples of physical (system) parameters are actual sensor noise, hand-over noise, number and relative positions and alignment of sensors, and frequency of sensor reports. Examples of tracking parameters are process noise (to give numerical stability), assumed sensor noise, initial covariance, and associations gate size. These parameters are discussed in more detail by VanDyke and Tomkins [1992] and in Section 5, and this list is not exhaustive.

Additional complications may be introduced by target terminations (missile disintegrates; collisions), target initiations (MIRV deployment; stage separation), maneuvering and other complications. Target initiations are included in the present study.

3.2 The TRC Simulation Package

The TRC simulation package¹ as received from NRL consisted of four² programs that ran in sequence. The first program generated the trajectories that describe the threat. The second program created simulated sensor reports based on the flight of those objects. This program did not do a sophisticated sensor simulation but merely added positional noise to the sensor reports using normally distributed random numbers. The third program was the Tracker Correlator. A significant portion of our work was concerned with making this program run efficiently in a massively parallel computing environment. This work began

---

1. Developed for the Naval Research Laboratory by D. H. Wagner and Associates and Ball Systems Engineering [Ball Systems Engineering Division and D. H. Wagner Associates, Inc., 1988a,b].

2. One could say there actually were five since the Sensor Report Generator consisted of a generator and a routine to format the reports for the TRC that required a sort to be run in between the two routines. The SORT routine would make six, but a sort program was not supplied.

when there was a significant concern with protection against a “phase one” threat and previous work included treating threats up to 57,000 objects. The fourth computer program was the MOE program which compared the tracking answers with the true position of the objects. As we received the TRC, the codes ran one after the other on a serial computer passing the information between them as data files. The volume of data required for even moderately sized threats (1000 - 2000 targets) motivated us to run the sensor report generation concurrently with the TRC.

An MOE calculation can take time at least comparable to the tracking simulation, so we created an option of also running the MOE calculation at the same time. The resulting program structure is sketched in Figure 2. The large shaded box encloses the three programs that run concurrently on the nCUBE/2.

4. Ramifications of the MPP Computing Environment

The massively-parallel processor environment was selected for this calculation for reasons of speed, and in view of the compatibility of the program structure with this environment. In particular, the 1024-processor nCUBE was used for the present trials. It also may be possible to conduct these calculations at the necessary speed on a single processor vector machine or even on a high-speed workstation.

Parallel implementation is a primary contribution of the present effort. The TRC was implemented, together with the sensor report generator, as shown in Figure 3. Details are presented by Tomkins and VanDyke [1991]. The key to the distributed architecture is the

Figure 3. TRC Parallel Implementation

use of gating nodes to manage clusters. Each post-boost vehicle is assigned to a worker node, which is responsible for processing sensor reports and updating tracks.

The full MOE calculation (see Fig. 2) was implemented in the "double node" architecture shown in Figure 4. For these calculations, gating nodes are also used in the MOE portions of the calculation. Each track is assigned to an MOE worker node, which is responsible for determining whether or not one of its Truths is suitable to associate with the assigned track. Just as in the TRC functionality for sensor data, Truths are gated to MOE worker nodes based on cluster properties (centroid and extent). As a result, each Truth may be given to more than one MOE worker node.

If Track to Truth matching is performed by associating the nearest Truth for each Track, this works quite efficiently. The decisions about which associations are best are made locally; no communication is needed between MOE worker nodes to determine the nearest Truth.

If Track to Truth matching is performed by associating the nearest Track for each Truth the process is more complex because of the distributed data that must be compared to determine the correct nearest Track. All worker nodes must report to the MOE gating nodes candidate Tracks for each Truth. The MOE gating nodes make the final decision of which associations are best. This is a global decision process in which significant communication between nodes is required to determine the associations.

Several complications arise in such a distributed data environment, or any environment which attempts to economize runtime through a scheme which does not compare all Tracks with all Truths. Aside from those complications just mentioned, extra care must be taken in managing redundant tracks and missed objects. Cluster overlaps must be monitored (these are time dependant). Finally, processor loads should be monitored to minimize bottleneck effects.

5. Sample problems

As noted in the Introduction, there is significant interest in threats involving up to 1,000 - 2,000 targets. Accordingly, we chose to conduct a set of runs with a 990 object threat. A launch of 99 vehicles is assumed, with 99 post-boost vehicles (PBV's) entering the mid-course phase. Each PBV spawns 9 additional objects, including multiple independently targetable re-entry vehicles (MIRV's) and decoys. This is labeled the 99-9 threat.

For this particular problem, 8 infrared sensors mounted on satellites monitor the threat. All satellites, PBV's, MIRV's and decoys are assumed to follow Kepler orbits with the J_2 correction applied [Vandyke and Tomkins, 1992]. It should be noted that the J_2 correction is inexpensive in segments (such as in TRC), but expensive in flyout (such as in truth calculation). Monitoring begins 248 seconds after launch and terminates 958 seconds after launch. Spawning of objects occurs over the time interval 245 - 655 seconds after launch, with individual PBV's spawning objects over times of approximately 80 seconds. (The threat described here is a test problem and is not meant to represent any real threat.)

Figure 4. "Double Node" implementation of MOE analysis

Table 1: Assumed and Actual Sensor Noise Parameters Used

		Assumed Noise (μrad)																
Actual Noise (μrad)		7	8	10	12	15	20	25	30	40	50	75	100	125	150	200	250	500
10		■	■	■	■	■	■	■			■							
50								■	■	■	■	■	■	■			■	
100											■		■		■	■	■	■

The primary MOEs investigated here were ghost tracks, missed objects, redundant tracks and unresolved objects. Actual and assumed sensor noise were varied as shown in Table 1.

This trial illustrates how performance of the TRC may be affected by the choice of parameters, and, through the feedback provided by the MOE, may be optimized.

The four measures of effectiveness are plotted against time in Figure 5 for an actual sensor noise of 10 μrad . Figures 6 and 7 give similar information for actual sensor noises of 50 and 100 μrad . Time averages of these MOEs are plotted versus assumed sensor noise in Figure 8 and tabulated in Table 2.

Table 2: Time averaged values of Measures of Effectiveness for TRC with 99-9 threat

Sensor Noise (μrad)		Redundant Tracks	Missed Objects	Ghost Tracks	Unresolved Objects
Actual	Assumed				
10.0	7.0	162.0	19.1	174.1	25.1
10.0	8.0	105.3	16.5	115.9	23.1
10.0	10.0	67.8	17.7	79.6	23.5
10.0	12.0	52.6	16.6	62.8	21.8
10.0	15.0	63.2	17.9	75.4	25.4
10.0	20.0	72.0	17.4	81.2	22.7
10.0	25.0	79.2	17.3	88.9	23.1
10.0	50.0	148.2	20.4	157.1	24.9
50.0	25.0	772.6	39.7	837.6	70.5
50.0	30.0	598.0	41.5	664.1	73.2
50.0	40.0	430.3	45.6	490.4	74.3
50.0	50.0	357.0	46.0	414.9	73.4
50.0	75.0	340.9	46.9	398.9	75.6
50.0	100.0	318.3	46.0	374.4	73.1
50.0	125.0	298.6	48.5	354.1	76.4
50.0	250.0	313.8	50.8	362.2	73.3
100.0	50.0	913.4	59.2	1040.1	119.7
100.0	100.0	522.4	69.2	638.2	120.1
100.0	150.0	499.5	73.2	610.9	122.3
100.0	200.0	473.3	75.8	583.5	124.2
100.0	250.0	471.6	76.4	580.7	124.9
100.0	500.0	478.8	90.8	579.8	130.7

Figure 5. Measures of Effectiveness for actual sensor noise of $10 \mu\text{rad}$.
Assumed sensor noise varied; see legend to right of plots.

Figure 6. Measures of Effectiveness for actual sensor noise of $50 \mu\text{rad}$.
Assumed sensor noise varied; see legend to right of plots.

Figure 7. Measures of Effectiveness for actual sensor noise of 100 μrad .
Assumed sensor noise varied; see legend to right of plots.

Figure 8. Time-averaged values of Measures of Effectiveness.

The actual sensor noise is a parameter determined by factors external to the tracker/correlator system. This depends on the hardware employed to monitor the targets. Ideally it should be as small as possible, but in practice programmatic constraints will place a lower bound on this parameter. The lowest value of 10 μ rad may be a practically achievable level. The assumed sensor noise, on the other hand, may be taken as any desired value. A very small value may be expected to cause correlation problems in gating and track updating because the algorithm will assume measurable significance for what in fact is random noise. On the other hand, a very large value may also allow the algorithm to make inappropriate associations.

The present trial shows that, for all three actual noise levels used, an assumed sensor noise less than the actual sensor noise causes increased redundant tracks and ghost tracks. This is consistent with the gating and correlation problems predicted above. Assuming a sensor noise greater than the actual sensor noise has a more ambiguous effect. An extreme case of assumed sensor noise equal to 5 $\times$ the actual sensor noise was tried for all three actual sensor noise levels. For the smallest actual sensor noise, this extreme case resulted in a marked increase in redundant tracks and ghost tracks. For the two larger sensor noises choices, however, the extreme case actually caused a late time decrease in the measures of effectiveness.

On the other hand, the numbers of missed objects and unresolved objects are not sensitive measures of effectiveness for sensor noise choices; these are almost independent of the assumed sensor noise chosen.

It is possible for the MOE algorithm to lose an object, which would then appear in the results as an missed object. We believe, however, that few, if any, of the missed objects reported here at late time are lost due to MOE algorithm errors. A brief analysis of several of the missed objects reported at late time for these runs suggests that they were lost by the TRC algorithm; *i.e.* that there were no tracks from the TRC algorithm that the MOE algorithm could reconcile with these objects. Objects which were reconciled with tracks were generally within ~ 100 m of these tracks, while the lost objects were quite distant. An analysis of the entire missed-object situation, especially during object spawning, would be very time-consuming and has not been attempted in the present study.

The most important parameters affecting the success of the TRC were described in some detail by VanDyke and Tomkins [1992]; for completeness these are summarized here:

1. Process noise is a parameter added to the elements of the covariance matrix to stabilize the calculation and to account for errors in the propagation model.
2. Assumed sensor noise is the uncertainty in the sensor readings assumed by TRC.
3. Actual sensor noise is the uncertainty in the actual sensor readings, or that uncertainty added to the correct readings (Gaussian distribution) by the report generator in an MOE calculation.
4. Initial covariance is the starting covariance (uncertainty) attached to a track at the handover from boost phase into mid-course phase.
5. Handover noise is the error in position and velocity of an object provided in hand-

over data.

6. Association gate size determines whether a sensor report will be considered for association to an existing track. In TRC this is defined in terms of the standard cost function [Blackman, 1986].
7. The number and relative position of sensors determines the amount of tracking data available for association.
8. The frequency of sensor reports affects the accuracy of tracking and how the other parameters affect tracking.

A complimentary study was discussed by VanDyke and Tomkins [1992]. This study, assuming a threat of a single missile, used as its measure of effectiveness the deviation of track from truth. Parameters varied included actual sensor noise, assumed sensor noise, process noise and handover noise.

Interestingly, for this study, the tracking seemed to monotonically improve as assumed sensor noise was increased over an interval similar to that evaluated in the present trials. This conclusion is artificial, however. In this study the handover noise was assumed to be zero (initial trajectory known exactly). Since an actual sensor noise was included in the calculations, allowing sensor reports to affect the track could only reduce the accuracy of the track. A large assumed sensor noise causes sensor reports to be given less weight in the calculation than does a small assumed sensor noise. Adding noise to the handover brought the results of this MOE study into qualitative agreement with the results of the present trials.

6. Conclusions

The effectiveness of a tracker/correlator algorithm for mid-course tracking may be expressed in terms of several parameters, including the number of ghost tracks, missed objects, redundant tracks and unresolved objects as a function of time, referred to as MOE parameters. Smaller values correspond to better algorithm performance for these parameters. The effectiveness of the algorithm is expected to depend on the choice of various input parameters, including the process noise, the assumed sensor noise, the initial covariance and the association gate size. As well, there are system parameters which must be added for any test of the effectiveness of the tracker/correlator algorithm, such as actual sensor noise, handover noise, the number and relative position of sensors and the frequency of sensor reports. Choosing unrealistic system parameters (such as zero handover noise) may cause unrealistic tracker/correlator performance. In particular, the dependence of the MOE parameters on input parameters may be unrealistic.

We have measured the MOE parameters for the tracker/correlator algorithm TRC, using a threat with 99 objects spawning to produce 990 objects. These measurements were made with several choices of actual sensor noise, and for each such choice, a variety of choices of assumed sensor noise. Other parameters were assigned values found to be reasonable in previous work.

Counts of redundant tracks and ghost tracks were minimized for choices of assumed sensor noise set equal to 1.2 to $2.5 \times$ the actual sensor noise (increasing with actual sensor noise). Decreasing the assumed sensor noise from these values caused more degradation of algorithm efficiency than did increasing it. Counts of missed objects and unresolved objects proved to be relatively insensitive to choice of assumed sensor noise.

The implementation of this MOE problem on an MIMD massively-parallel computer has several advantages, notably a marked speedup in execution of the tracker/correlator and the capability of running the MOE concurrently with the tracker/correlator. On the other hand, extra effort is required for gating, matching tracks between clusters, and assuring a load balancing among nodes to avoid bottlenecking.

References

- Allen, T. G., Multiple hypothesis tracking algorithms for massively parallel computers, *Signal and Data Processing of Small Targets 1992*, O. E. Drummond (ed.), Proc. SPIE 1698, pp. 442-456, 1992.
- Ball Systems Engineering Division and D. H. Wagner Associates Inc., SDI tracker/correlator algorithm design document, second edition, R-061-88, Naval Research Laboratory, July, 1988
- Ball Systems Engineering Division and D. H. Wagner Associates Inc., SDI tracker/correlator algorithm implementation document, second edition, R-063-88, Naval Research Laboratory, July, 1988
- Blackman, S. S., Multiple-target tracking with radar applications, Artech House, Inc., 1986.
- Drummond, O. E. and B. E. Fridling, Ambiguities in evaluating performance of multiple target tracking algorithms, *Signal and Data Processing of Small Targets 1992*, O. E. Drummond (ed.), Proc. SPIE 1698, pp. 326-337, 1992.
- Roecker, J. A., Track monitoring with single and multiple 2-D passive sensors, *Signal and Data Processing of Small Targets 1992*, O. E. Drummond (ed.), Proc. SPIE 1698, pp. 416-424, 1992.
- Tomkins, J. L. and VanDyke, J. P., "Massively parallel computing and the mid-course tracking problem," *Proceedings of the Supercomputing '91 Conference*, Albuquerque, New Mexico, November 18-22, 1991, pp. 796 - 804.
- Vandyke, J. P. and J. L. Tomkins, The impact of the J_2 gravitational term on SDI mid-course tracking, Sandia National Laboratories Report, SAND92-0490, 1992.

Distribution

External

Dr. Kurt Askin
Naval Research Laboratory
ATTN: Code 5570
4555 Overlook Avenue, SW
Washington, DC 20375-5000

Dr. Barry Belkin
Daniel H. Wagner Associates
Station Square Two
Paoli, PA 19301

Dr. Jay Boris
Naval Research Laboratory
Code 4400
4555 Overlook Avenue, SW
Washington, DC 20375-5000

Dr. James Boyle
Argonne National Laboratory
Math and Computer Science Division
9700 South Cass Avenue
Argonne, IL 60439-4844

Dr. Donald Brand
NTBIC/Geodynamics
c/o Martin Marietta ISG
NTB Division
Mail Stop N8930
Falcon AFB, CO 80912-5000

Dr. Hal Camp
ARES Corporation
1111 North 19th Street
Suite 305
Arlington, VA 22209

Dr. Prasanta Das
The Analytic Sciences Corporation
1101 Wilson Boulevard
Suite 1500
Arlington, VA 22209

Dr. Oliver Drummond
General Dynamics Air Defense System Division
MZ 601-79
P.O. Box 5080
Ontario, CA 91761-1085

Dr. Keh-Ping Dunn
MIT Lincoln Laboratory
P.O. Box 76
Room KB-252C
Lexington, MA 02173-9108

Dr. Gabriel Frenkel
Institute for Defense Analyses
1801 North Beauregard Street
Alexandria, VA 22311

Dr. Barry Fridling
Institute for Defense Analyses
1801 North Beauregard Street
Alexandria, VA 22311

Dr. James P. Hardy
Geodynamics
National Test Bed Program
Mail Stop N8930
Falcon AFB, CO 80912-5000

Mr. Daniel Holtzman
Vanguard Research, Inc.
10306 Eaton Place
Suite 450
Fairfax, VA 22030-2201

Dr. David P. Kierstead
Daniel H. Wagner Associates
450 Maple Avenue, East
Suite 206
Vienna, VA 22180

DR. Ronald Kolbe
Naval Research Laboratory
Code 4440
4555 Overlook Avenue, SW
Washington, DC 20375-5000

Dr. Michael Kovacich
Lockheed Missiles and Space Company
1111 Lockheed Way
Building 554
Sunnyvale, CA 94089-3504

Mr. Al Perrella
Institute for Defense Analyses / POET
1225 Jefferson Davis Highway
Suite 300
Arlington, VA 22202

Dr. Leslie Pierre (5)
SDIO/SDB
Room 1E149
The Pentagon
Washington, DC 20301-7100

Ms. Tina Powell
Vanguard Research, Inc.
10306 Eaton Place
Suite 450
Fairfax, VA 22030-2201

Mr. Stephen Rhodes
Advanced Systems Architectures, Ltd.
North Block, Bentley Hall
Blacknest, Alton, Hants
GU34 4PU
United Kingdom

Mr. Steve Risner
USASDC
ATTN: CSSD-SA-BT
P. O. Box 1500
Huntsville, AL 35807

Dr. James Sanderson
Los Alamos National Laboratory
P. O. Box 1663
MEE-10, MS K488
Los Alamos, NM 87545

Dr. Patrick Shea
TRW, Inc.
One Space Park Drive
Building R2, Room 1028
Redondo Beach, CA 90278

Mr. William Stonestreet
Alphatech, Inc.
50 Mall Road
Burlington, MA 01803-4537

LtCol James Sweeder
SDIO/SDA
Room 1E149
The Pentagon
Washington, DC 20301-7100

Dr. Robert Washburn
Alphatech, Inc.
50 Mall Road
Burlington, MA 01803-4537

Internal

MS 0360	Albert R. C. Westwood, 1000	
MS 0321	E. H. Barsis, 1400	
MS 1109	A. L. Hale, 1424	
MS 1109	J. L. Tomkins, 1424	(10)
MS 1109	J. P. Vandyke, 1424	(10)
MS 0441	P. L. Stanton, 1425	
MS 0820	P. Yarrington, 1432	
MS 0821	M. D. Furnish, 1433	(10)
MS 9018	Central Technical Files, 8523-2	
MS 0899	Technical Library, 13414	(5)
MS 0619	Print Media, 12615	
MS 0100	Document Processing, 7613-2	(2)