COMPARISON OF LINEAR AND NON-LINEAR TRACKERS

Report to

Naval Research Laboratory

by

Lawrence D. Stone
Thomas M. Kratzke
Metron, Inc.

16 December 1991

TABLE OF CONTENTS

PREFACE

This work was performed under contact N00014-89-C-2341. Technical direction was provided by Dr. Thomas Mifflin.

SUMMARY

This report presents the results of a series of tests designed to compare the performance of the discrete non-linear tracker, Nodestar, to that of an extended Kalman filter in a variety of situations involving the use of line-of-bearing detections to track a submarine. The extended Kalman filter used in this comparison is MTST (Maneuvering Target Statistical Tracker). MTST is used in a number of Navy systems including the OBU tracker and NTSA's TIMS/MRS track reconstruction system.

Thirty cases were run. In each case, there were three fixed passive arrays and a target track that was chosen in a random fashion. Detection of the target by the arrays was simulated using a λ-σ process for acoustic fluctuations. When a detection occurred, the simulation calculated the true bearing of the target to detecting sensor and then added a random error. Each case lasted for 20 hours of simulated time. For each case the file of detections and noisy bearings was recorded. The same file was fed to Nodestar and MTST. Every 15 minutes during the run, each tracker produced an estimate of the target's location in terms of a probability distribution. These estimates were recorded and used to calculate MOEs that measure the performance of the two trackers.

S.1 Description of MOEs

Both Nodestar and MTST are statistical trackers in the sense that their estimates of target position are in terms of probability distributions. MTST's output is displayed as ellipses. The ellipses represent bivariate normal distributions for target location and are the contours of the 86% containment region. Nodestar's output is a discrete probability distribution on target state. Each cell in the distribution is color coded to indicate its probability.

We used three MOEs:

AOU Size. This is the area of the smallest region containing 86% of the probability distribution on target location. For MTST this is the area of the 86% (2-σ) ellipse.

Mean Missed Distance (MMD). For this measure we must know the target's actual position. We then calculate the mean squared distance of the target's actual position from the probability distribution on target location. The root mean squared distance is defined as the mean missed distance (MMD). MMD is the generalization to probability distributions of the mean missed distance that one would calculate if the tracker gave a point estimate for target position.

Accuracy. We measure the accuracy of the probability distributions produced by a statistical tracker in terms of its containment regions. As an example take the 86% containment region for MTST (i.e., the ellipses). Suppose we plotted the actual target position at 100 times and compared it to the 86% region at those times. We want the target's position to fall inside the region 86% of the time and outside the remaining 14%. Similarly if looked at the 50% region, we would want the target to be in 50% of the time and out the other 50%. If this is true, then our distribution is giving us an accurate representation of the degree to which we have localized the target.

Accuracy is a measure of the extent to which a tracker's containment regions are an accurate representation of its knowledge of the target's location. Accuracy varies from 100% to 0%.

In making our comparison, we calculated the AOU size and MMD at each time the trackers made an estimate (i.e, 81 times for each case) and computed the average AOU size and MMD as our comparison statistics. The accuracy MOE is computed for an entire case. It is based on the distributions produced at the 81 times during the run.

S.2 Robustness Testing

Both Nodestar and MTST require the user to make probabilistic assumptions about the target's motion. Since Nodestar uses detection / no-detection information from the arrays to help estimate the target's location, it requires an estimate of the Figure of Merit (FOM) for the target's primary (i.e., most detectable) frequency and the propagation loss curve for that frequency. This acoustic information is necessary for the detection / no detection likelihood functions calculated by Nodestar. We investigated the robustness of the trackers' performance when there is a mismatch between the assumptions used by the tracker (i.e., the motion and

acoustic assumptions) and those used in the simulation that generated the file of detections fed into the trackers.

S.3 Test Scenarios

We used the following scenarios in our tests.

Target Motion Models. For the target motion in the simulator, we used three basic types, patrolling, constant-course-and-speed, and evading.

Both Nodestar and MTST used one fixed motion model for all of the 30 test cases. The Nodestar motion model assumed that the target's heading was uniform [0, 360] and its speed distribution uniform over [5kts, 20kts]. The target course changes were modeled as taking place at exponentially distributed times with mean time 1 hour between course changes.

The motion model in MTST is an Integrated Ornstein-Uhlenbeck process. The choice of the parameter values for this motion model was made in conformance with the recommendations in reference [a]. The objective was to match the motion assumptions used by Nodestar as closely as possible.

Acoustic Assumptions. The test considered three acoustic environments, surface duct; direct path and bottom bounce; and convergence zone.

Nodestar requires that the user specify a propagation loss curve and an FOM for the target. To test the robustness of Nodestar to mismatches in the propagation loss curve and the FOM, we considered the following cases:

1. Correct transmission loss curve, correct estimate of figure of merit.
2. Correct transmission loss curve, high estimate of figure of merit.
3. Correct transmission loss curve, low estimate of figure of merit.
4. Incorrect transmission loss curve, correct estimate of figure of merit.

We ran one test for each combination of assumptions listed above. This produced 3 motion models × 3 propagation loss curves × 3 FOM assumptions + 3 mismatched propagation loss curves = 30 runs.

Each run was given a 3-letter label that describes it. Table S.1 describes the meaning of the 3-letter label. The first letter identifies the acoustic environment; C stands for convergence zone, S for surface duct, and D for direct path. The second letter identifies the motion of the

simulated target; P stands for patroller, E stands for evader, and C stands for constant course and speed. The third letter identifies the accuracy of the acoustic information; C stands for correct information, H stands for a high estimate of the FOM, L stands for a low estimate of the FOM, and I stands for an incorrect transmission loss assumption by Nodestar.

Table S.1

RUN LABELS

First Letter	Second Letter	Third Letter
Environment	Target Motion	Mismatch
C = CZ	P = Patroller	C = Correct Acoustic Assumptions
D = Direct Path, BB	C = Constant Course and Speed	H = High estimate of FOM
S = Surface Duct	E = Evader	L = Low estimate of FOM
		I = Incorrect Prop Loss

S.3 Results of Comparison

Table S.2 gives a summary of the results of the 30 runs. Each line of the table shows the values of the three MOE's (AOU size, mean missed distance (MMD), and accuracy) that were computed for Nodestar and MTST for the indicated run. Each run covered a period of 20 hours. Every 15 minutes each tracker produced an estimate of the target's location for a total of 81 times at which estimates were produced (including the initial estimate at time 0). The better of the two numbers in each category for a run is labeled with an asterisk. A quick glance shows that Nodestar wins most of the comparisons. The columns labeled G and H give the number of times at which contact was gained and held respectively during a run. The sum of the gains and holds yields the total number of contacts made by all three arrays during the run.

The first row of Table S.2 shows the results for the case of a convergence zone environment, constant-course-and-speed target, and correct estimates by the Nodestar of the FOM and propagation loss curve. Note that Nodestar has a lower average AOU size, smaller mean missed distance (MMD), and higher accuracy than MTST. The next two lines show the

results for two cases with the same target motion and environmental characteristics as the first one, but with Nodestar assuming that the FOM is one standard deviation higher and lower respectively than it actually is in the simulation. *Even with this mismatch in acoustic assumptions, Nodestar wins in all three categories.*

One would expect Nodestar to outperform a Kalman filter such as MTST in a convergence zone environment. The results in Table S.2 bear that out. In 27 comparisons, there are only three instances in which MTST has a better MOE than Nodestar. Overall, Nodestar is clearly superior.

In the case of a direct path or ducting environment one might expect that Nodestar's edge over MTST would diminish because the acoustic information in this environment would not be as valuable as in a convergence zone environment. *Surprisingly, the results in Table S.2 show that Nodestar continues to outperform MTST in direct path and surface duct environments even in the face of mis-estimates of FOM.*

Examining the three runs (DCI, DEI, and DPI) in which the propagation loss curve was convergence zone but Nodestar thought it was direct path, we see that Nodestar bettered MTST in 8 out of the 9 MOE comparisons.

Figures S-1 through S-3 show a graphical presentation of the comparisons in the table. The comparison of AOU sizes is shown in Figure S-1. The cases are labeled along the horizontal axis. For each case there is a bar showing the difference between the AOU size for Nodestar and MTST. The height of the bar indicates the magnitude of the difference. A black bar indicates that Nodestar had a smaller AOU size than MTST. A white one indicates that MTST had a smaller AOU size. One can see that Nodestar beat MTST in all cases except one.

Figure S-2 shows the mean missed distance comparison. Nodestar beats MTST in all cases except 2. Figure S-3 shows the accuracy comparison. Nodestar wins 21 out of 30 of these comparisons.

S.4 Conclusions

The obvious conclusion is that Nodestar performs much better than MTST in favorable situations (e.g., convergence zone environments with correct acoustic estimates) and continues to outperform MTST in cases where Nodestar is given a poor estimate of FOM or even the wrong propagation loss curve. This conclusion was tested statistically using a Wilcoxen test

for paired observations. This test showed that Nodestar performed significantly better than MTST for all three MOEs.

One might ask whether the difference between Nodestar and MTST could be attributed to the possibility that MTST is not a good implementation of a Kalman filter. In our minds the answer to this question is clearly no. MTST is one of the best Kalman filters for submarine tracking. It has a motion model that provides sensible velocity distributions and it handles non-linear information in a reasonable fashion given the linear constraints imposed by the Kalman methodology.

When the output from Nodestar and MTST are viewed side by side at each time step, the reasons that MTST performs so poorly compared to Nodestar become clear. First, the lack of range information in the bearing detections used by MTST makes it difficult for MTST to localize the target. The linear approximations required to incorporate line-of-bearing detections cause the target distribution to get drawn into the sensor in a sort of fatal attraction. The inability to use negative information allows the target distribution to expand into regions near the sensor over long periods when no contact is obtained. All of these problems will be shared by any Kalman filter. They are not specific to MTST.

Table S.2

SUMMARY RESULTS

Run	AOU (nm^2)		MMD (nm)		Accuracy (%)			
	Nodestar	MTST	Nodestar	MTST	Nodestar	MTST	G	H
CCC	2,839*	4,880	36.83*	64.76	69.76*	27.62	4	13
CCH	2,354*	4,158	33.47*	54.91	79.91*	33.36	5	16
CCL	946*	1,122	23.41*	28.74	70.19*	45.18	10	49
CEC	1,571*	2,530	25.62*	33.39	67.19*	65.17	6	52
CEH	2,874*	5,822	41.61*	69.11	52.34	53.88*	7	8
CEL	2,908*	4,641	46.84*	46.94	46.38	59.28*	4	23
CPC	4,597*	7,584	41.07*	45.31	67.58	90.45*	3	9
CPH	4,017*	7,247	32.59*	49.24	88.3*	65.90	3	19
CPL	11,080*	16,860	56.83*	65.19	84.48*	81.9	2	0
DCC	1,057*	1,743	21.73*	48.21	50.7*	30.39	5	32
DCH	4,852*	8,849	66.21*	74.38	19.04	41.13*	2	7
DCL	1,911*	3,620	36.07*	58.60	25.61	27.99*	3	26
DCI	1,482*	2,595	21.45*	50.62	63.57*	36.42	6	25
DEC	3,705*	6,688	40.86*	63.76	71.65*	61.95	3	5
DEH	2,012*	3,697	24.07*	32.63	71.41	79.19*	4	29
DEL	3,768*	6,147	66.64*	96.15	30.05*	25.37	2	13
DEI	4,664*	8,129	66.68*	70.99	27.02	44.83*	4	5
DPC	898*	1,322	21.55*	28.90	70.83*	68.99	9	24
DPH	2,155*	3,831	36.98*	41.28	41.42	70.93*	6	16
DPI	2,670*	4,706	26.82*	39.80	72.15*	61.60	4	26
DPL	6,680*	10,940	53.20*	60.94	62.53*	58.07	2	0
SCC	5,668*	8,162	72.26	63.92*	39.96	54.45*	7	3
SCH	1,224*	1,529	22.34*	39.37	83.21*	40.36	8	30
SCL	5,755*	9,273	70.14*	75.94	53.64*	43.45	3	4
SEC	24,920	21,800*	181.50	133.10*	9.77	32.00*	0	0
SEH	3,275*	5,139	30.06*	65.81	89.13*	50.01	4	22
SEL	3,160*	4,410	41.55*	45.08	65.58*	65.25	5	16
SPC	1,750*	2,587	27.43*	34.01	71.12*	70.91	8	13
SPH	1,639*	2,310	29.66*	40.33	62.48*	50.30	6	20
SPL	860*	1,261	15.97*	33.90	89.47*	43.12	11	22

* indicates the winner.

LABELS

Environment	Target Motion	Mismatch
C = CZ	P = Patroller	C = Correct Acoustic Assumptions
D = Direct Path, BB	C = Const Crse & Spd	H = High estimate of FOM
S = Surface Duct	E = Evader	L = Low estimate of FOM
		I = Incorrect Prop Loss

Figure S-1

AOU COMPARISON

Nodestar vs MTST

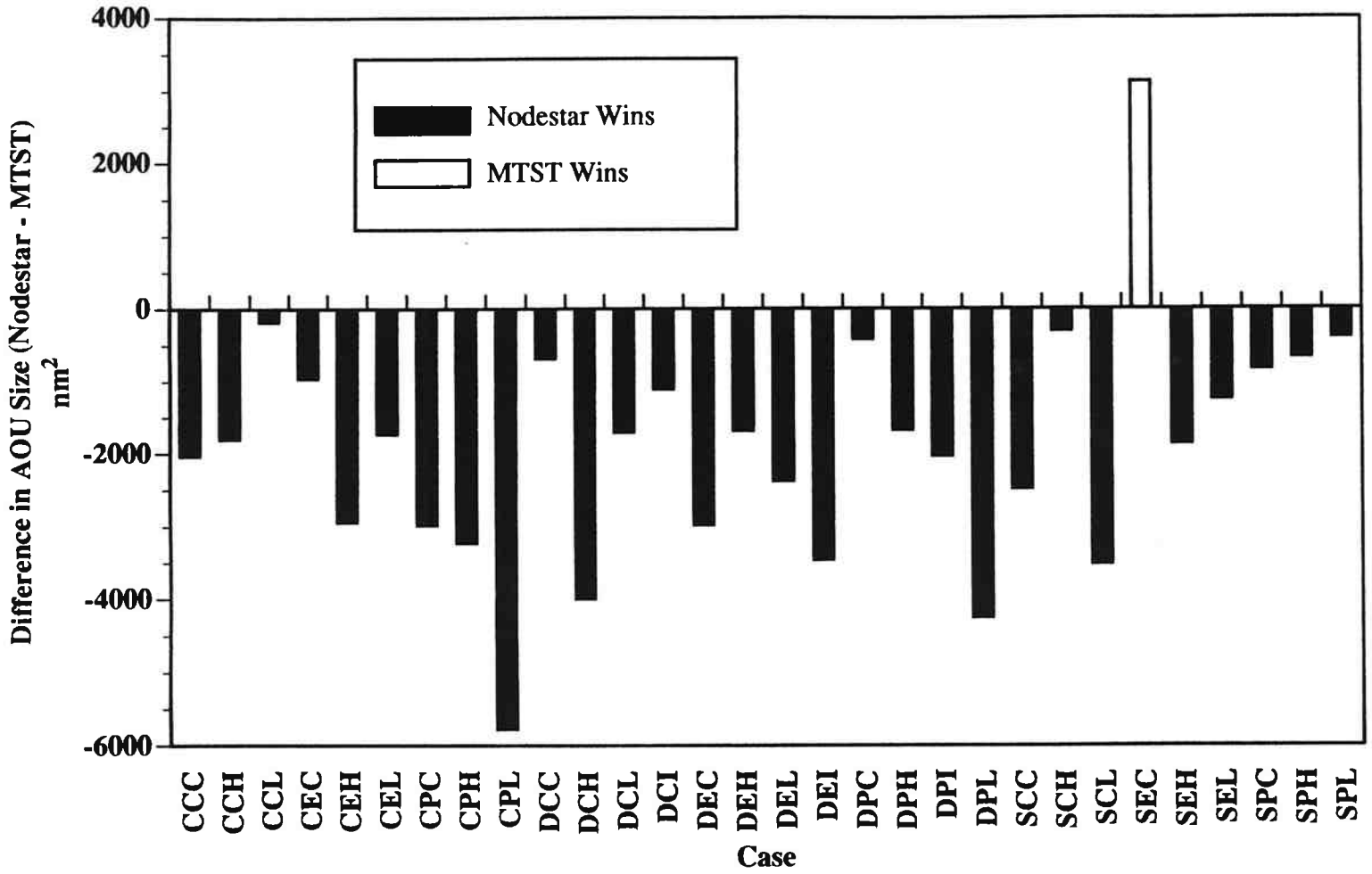


Figure S-2

MEAN MISSED DISTANCE COMPARISON

Nodestar vs MTST

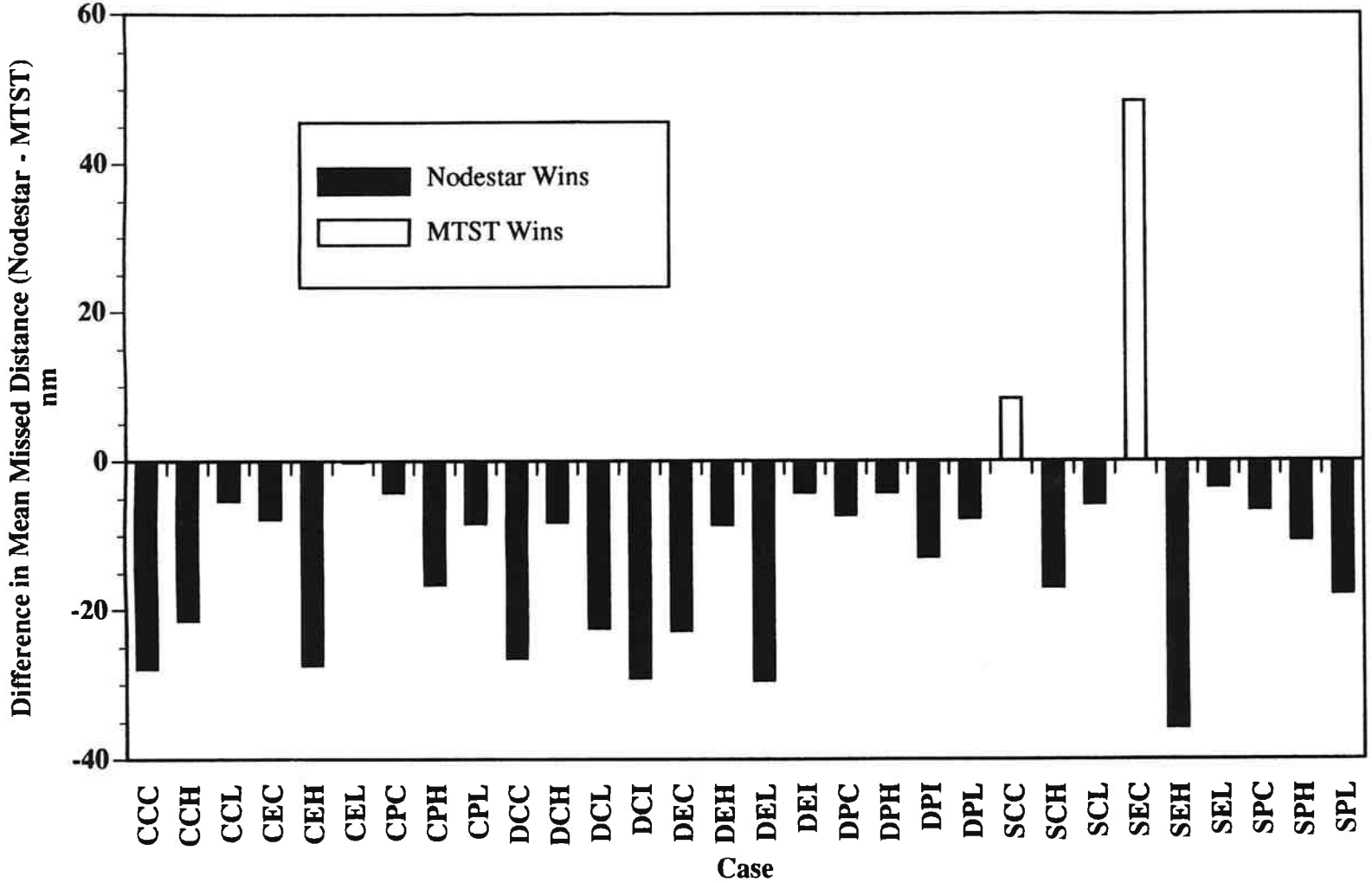
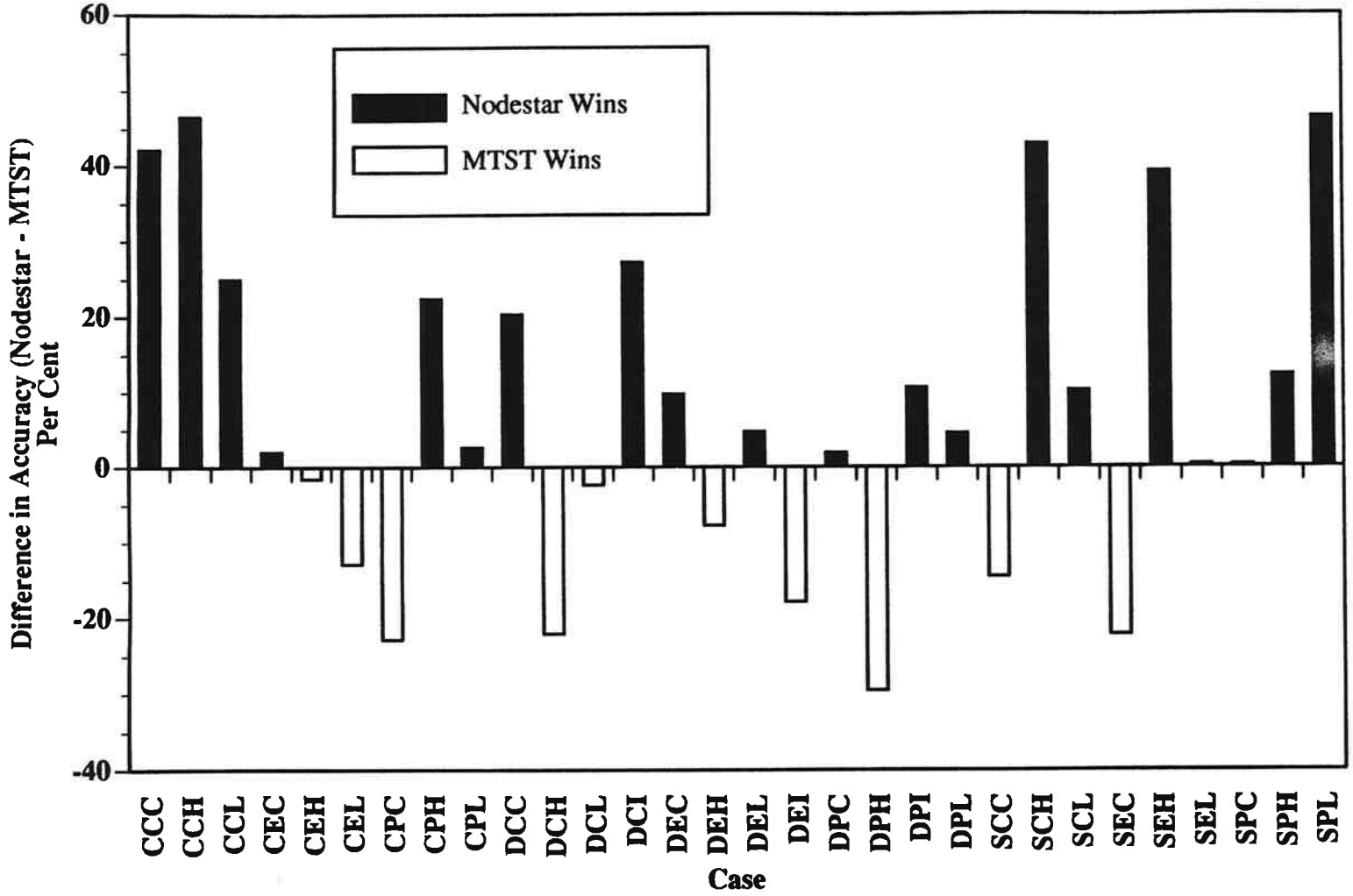


Figure S-3

ACCURACY COMPARISON

Nodestar vs MTST



INTRODUCTION

For many years the Navy has used Kalman filters to track submarines using line-of-bearing data obtained from passive acoustic arrays. Kalman filters have always had difficulty with this tracking problem because they assume that the relationship between the observation and the target state is linear. Line-of-bearing information is non-linear. In order to accommodate this information in a Kalman filter, one typically approximates the non-linear relation by a linear one. The result is called an extended Kalman filter. Use of an extended Kalman filter can lead to serious problems such as range collapse, back bearing solutions, and filter divergence.

Another problem with Kalman filters is the requirement that measurement errors be Gaussian. This requirement along with the linearity assumption, means that the estimates of target state produced by a Kalman filter are always Gaussian and unimodal. If one obtains a line-of-bearing detection in a convergence environment, one has to approximate the multimodal distribution that represents the detection by a unimodal Gaussian distribution. This is clearly misleading and will produce poor estimates of target location.

Metron has developed a discrete non-linear tracking system called Nodestar. Nodestar (previously called LRT or the Likelihood Ratio Tracker) is capable of incorporating a wide variety of information, e.g., lines of bearing, detection no-detection information, elliptical contact information, land avoidance information, and many others without assuming linear relationships or Gaussian measurement errors.

One may view Kalman filtering as a special case of the non-linear filtering performed by Nodestar, in the sense that if one limited Nodestar to linear relationships between the measurement and the target state and to Gaussian measurement errors, Nodestar would produce the same results as a Kalman filter (in discrete form). Thus Nodestar can do everything that a Kalman filter can do and much more.

In this report we present the results of tests comparing the performance of Nodestar to that of an extended Kalman filter in a variety of situations involving the use of line-of-bearing detections to track a submarine. The extended Kalman filter used in this comparison is MTST (Maneuvering Target Statistical Tracker). (See reference [a]). MTST is used in a number of Navy systems including the OBU tracker and NTSA's TIMS/MRS track reconstruction system.

Thirty cases were examined. In each case a simulation was run to produce a series of detections on a specified target track. When a detection occurred the actual bearing to the target from the sensor was calculated. Then an error term was drawn from a normal distribution and added to the actual bearing. The resulting noisy bearing was recorded on a file. The noisy bearings simulate actual bearing calls made by the sensor. The error term is added to account for the measurement error in the sensor. For each of the 36 cases, this file of noisy bearings was fed into Nodestar and into a Kalman filter (MTST). The performance of the two trackers was compared in terms of AOU size, mean missed distance, and accuracy.

The results of these tests show that Nodestar does a much better job of tracking submarines using information from passive sonar arrays than does a Kalman filter. Nodestar performs significantly better in terms of AOU size, mean missed distance, and accuracy. The only penalty that one pays when using Nodestar is that it requires substantially more computer power than a Kalman filter. However, with the small, inexpensive, and powerful computer hardware that is available today, one can easily meet Nodestar's computational requirements.

CHAPTER I

LINEAR AND NON-LINEAR TRACKERS

This chapter defines linear and non-linear trackers and provides a brief description of Nodestar.

1.1 Definition of Linear and Non-Linear Trackers

Jazwinski, in his classic text on filtering theory (see reference [b], chapters 5 - 7), defines and discusses linear and non-linear filters (i.e., trackers). For the purpose of this memorandum, we will paraphrase, in words, his definitions. A *linear tracker* is one in which the target's motion follows a linear (stochastic) differential equation and where the measurements (e.g., contacts) are linear functions of the target state. If, in addition, the measurement errors are Gaussian, then we have a *linear-Gaussian tracker*. Kalman filters are examples of linear-Gaussian trackers. Extended Kalman filters replace non-linear measurement relationships with local linear approximations in order to retain the computational simplicity of the linear-Gaussian tracker. A linear-Gaussian tracker always produces Gaussian distributions as estimates of the target state.

A *non-linear tracker* is one where the target's motion is allowed to follow a non-linear (stochastic) differential equation and the relationship between the measurements and the target state can be non-linear. If, in addition, the measurement errors are allowed to be non-Gaussian, then we have a *non-linear, non-Gaussian tracker*. Nodestar is a discrete-time, discrete-state space, non-linear, non-Gaussian tracker.

Kalman filters are well known and there are many references that describe them in detail, for example, reference [b]. The Nodestar methodology is less well known, so we shall describe it in the next section.

1.2 Basic Nodestar Methodology

Nodestar is a discrete, recursive, non-linear filter. It uses a Bayesian approach to compute estimates of target state (position, velocity, depth) in terms of probability distributions. Nodestar has the following components:

1. *Target Motion Model.* This is a stochastic process, i.e., a discrete time and space Markov chain, that models in a probabilistic sense the target's motion. The state space X of this process is five dimensional (2 position, 2 velocity, 1 depth). The parameters of this process can be chosen by the user to fit historical target motion behavior (e.g., speed-depth profiles) or subjective estimates of the target behavior based on the present tactical situation. Nodestar is initialized with a prior distribution on target state at some time $t = 0$.

2. *A Set of Likelihood Functions.* Nodestar has one likelihood function for each sensor modeled and type of information obtained from that sensor. As an example consider a towed array and consider two types of information that can be obtained from the array: (1) detection/no detection and (2) an estimated bearing (including the mirror bearing) for the target given a detection has occurred. For each type of information Nodestar calculates a likelihood function L. Suppose there are two towed array sensors, sensor 1 and 2. Suppose sensor 1 detects a target at time t and the called bearing is B with mirror bearing B'. Let L_1 be the detection likelihood function and L_2 be the bearing likelihood function. Then

$$L_1(x,t) = \Pr\{\text{sensor 1 detects at time } t \mid \text{target state} = x\}$$

$$L_2(x,t) = \Pr\{\text{sensor 1 calls bearings B and B' } \mid \text{target state} = x\} \quad \text{for } x \in X.$$

Suppose sensor 2 fails to detect the target at time t. Then Nodestar calculates a likelihood function, say L_3, for this information as follows:

$$L_3(x,t) = \Pr\{\text{sensor 2 fails to detect at time } t \mid \text{target state} = x\} \quad \text{for } x \in X.$$

The calculation of L_1 or L_3 can make use of a sensor detection model at any desired level of detail and fidelity. Present versions of Nodestar use the figure of merit and propagation loss curve for a selected primary frequency to calculate probability of detection or non-detection.

Computation of L_1, L_2, and L_3 is discussed in detail in reference [c]. This methodology can easily take advantage of range dependent propagation loss curves.

The motion model and likelihood functions are employed in the basic recursion used to produce estimates of target state. The recursion has the following three steps.

1. Motion Update. Let P_{t-1} be the distribution on target state at time t-1. The motion update applies the Markov transition probabilities (which may be space and time dependent) to produce P_t^-, the motion updated target state distribution at time t. Symbolically we write

$$P_t^- \leftarrow P_{t-1}.$$

2. Information Update (fusion). Let L_i i=1,..., I be the likelihood functions corresponding to the information obtained at time t. Compute

$$P_t^+(x) = P_t^-(x) \prod_{i=1}^{I} L_i(x,t) \qquad \text{for } x \in X.$$

3. Normalization. Renormalize P_t^+ so that it is a probability distribution, P_t, i.e.,

$$P_t(x) = \frac{P_t^+(x)}{\sum_{y \in X} P_t^+(y)} \qquad \text{for } x \in X.$$

The resulting function, P_t is the posterior distribution on target state at time t.

Dynamic Windowing. Nodestar contains an important innovation called *dynamic windowing*. All likelihood and probability computations in Nodestar take place inside a window, e.g., a 50 by 50 grid of spatial cells. This window is automatically translated to be centered at the mean of the Nodestar spatial distribution (i.e., the two-dimensional marginal distribution on target location). It is also increased or decreased in size as the Nodestar probability distribution increases or decreases in size. However, the number of cells always stays the same. The result is that as the size of the distribution shrinks, the size of the spatial cells automatically decreases and the resolution of the target state estimate increases. This feature allows Nodestar to handle problems that start as large scale searches and proceed down

to fine scale localization while providing the appropriate resolution and maintaining a relatively constant computational load. Dynamic windowing is applied to the velocity grid as well as the spatial grid.

1.3. Advantages of Nodestar

Let us make some observations. First the product in step (2) above assumes that the probabilities represented by the likelihood functions are independent. This is very likely to be true for different sensor types, e.g., acoustic and nonacoustic sensors, and for sensors located on different platforms. The independence assumptions are not so well satisfied for nearby sensors detecting on the same frequency, e.g., closely located sonobuoys. Thus Nodestar is an excellent tool for fusing diverse types of information. Let us discuss a point of terminology. In our lexicon, we use *correlation* for the process of assigning or associating information (e.g., contacts) with targets. *Fusion* is the process of combining all the information correlated with a target to produce an estimate of the target's state.

Another restriction in Nodestar is that in order to preserve the recursive nature of the algorithm, the probability of observing a piece of information (e.g., detection / no detection) must depend only on the target's state at time t. Since the state space includes position, velocity, and depth, this is usually not a problem. For example, Doppler information is readily incorporated because of the velocity component of the state space.

The possibility of a target maneuver is naturally handled by the stochastic process motion model. At each time there is some probability of the target maneuvering (this probability is controlled by the parameters that the user selects for the motion model). If the information is consistent with a maneuver (i.e., a change in target velocity), the mass of the probability distribution will shift from being concentrated near the old velocity to the new. The stronger the information, the faster the distribution will shift.

In Nodestar there is no requirement that the relationship between the target's state and the observation be linear. The Nodestar likelihood functions can handle observation errors that are non-Gaussian and multimodal.

Because Nodestar estimates the target state in terms of a probability distribution (which is usually displayed to the user in terms of a color probability map showing the marginal distribution on target position), it automatically shows the quality of the estimate and displays for the user the ambiguities in the estimates. The tightness of the probability distributions

shows the user the quality of the estimates. Ambiguities are represented by multiple modes. Let us summarize the strengths of Nodestar :

1. Handles maneuvering targets.

2. Fuses all information about the target, positive, negative, acoustic, and nonacoustic.

3. Incorporates subjective tactical information through the prior distribution on target states and specification of motion model parameters.

4. Allows for non-linear relationships between target state and observation.

5. Properly handles non-Gaussian and ambiguous (multimodal) information.

6. Provides target state estimates in terms of probability distributions which show the quality of the estimates and display ambiguities when appropriate.

The Kalman filter can be obtained as a special case of the Nodestar recursive filtering approach. In order to obtain this special case, one must limit all distributions to be Gaussian, all relationships to be linear, and use only positive (contact) information. For the modest computational resources available in the 60's, 70's, and early 80's, these limitations made sense because the Kalman filter requires little computational power. However, with the availability of inexpensive, powerful computers, these limitations are no longer appropriate.

CHAPTER II

COMPARISON METRICS FOR STATISTICAL TRACKERS

In this chapter, we present three measures for evaluating the accuracy of statistical trackers. A statistical tracker is one that produces a probability distribution for the estimate of target state as opposed to a point estimate. Section 2.1 discusses measures for evaluating statistical trackers and Sections 2.2 - 2.4 define the measures that we used to evaluate Nodestar and MTST.

2.1 Measures for Evaluating Statistical Trackers

We consider three measures of the accuracy of a statistical tracker. The second and third measures rely on knowing the actual target positions at the times at which tracker estimates are produced.

The first measure is the AOU size, the area of the smallest region having 86% probability of containing the target. For a linear-Gaussian tracker this is the area of the 2-σ uncertainty ellipse about the mean target position.

The second measure is the expected missed distance from the probability distribution to the actual target position. This measure is a straightforward generalization of the missed distance metric that one would use to measure the accuracy of a tracker that provides point estimates of target location.

The third measure is a quantile or containment measure which we call accuracy. To motivate this measure, consider a classical Kalman filter tracker that provides estimates in terms

of a mean and an elliptical probability region about that mean. The ellipse plus the mean define a multivariate normal distribution for the estimate of the target state. The Bayes estimator for this distribution is the mean. Suppose this tracker produces a series of estimates which are displayed by showing the mean of the estimates and the 50% ellipse associated with each estimate. Suppose we plot the actual target positions at the time of each estimate and count the fraction of times the target position is inside the ellipse. If the probability distributions for the target state estimates are accurate, the target should be inside the 50% ellipse 50% of the time. By adjusting the size of the ellipses, we can produce P% ellipses for any P such that $0 \leq P < 100$. For each P, we would expect the target to fall inside the P% ellipse P% of the time.

If the target is in the 50% ellipse less than 50% of the time, the tracker is being optimistic about the degree of target localization that has been achieved. Such optimistic estimates might cause one to fire a weapon prematurely because he thought he had achieved sufficient localization to justify firing. Similarly, if the target is inside the 50% ellipse more than 50% of the time, the tracker is being pessimistic about the degree of localization achieved. This might cause one to hold fire longer than necessary to achieve the degree of success desired. Such delay may result in loss of track altogether or an unnecessary exposure to risk of counter-attack.

An equivalent way of measuring these containment probabilities for the normal distributions discussed above is to consider the actual target position X_t at time t and compute the containment probability q_t of the ellipse on which X_t lies. If the distribution is an accurate representation of probability distribution on target location, then q_t will have a uniform distribution on [0,1]. If we compute q_t for a series of times $t = 1,\ldots,T$, and if the error in the mean position compared to the actual target position at time t is independent of the error at all other times, then the sequence $\{q_t\ ;\ t = 1,\ldots,T\}$ will be distributed as a sequence of independent draws from a uniform distribution on [0,1]. The hypothesis that $\{q_t\ ;\ t = 1,\ldots,T\}$ is such a sequence can be tested using a Kolmogorov-Smirnov (K-S) test. (See reference [d].) The measure accuracy that we define below quantifies how closely the sequence $\{q_t\ ;\ t = 1,\ldots,T\}$ fits the assumption of being independent draws from a uniform distribution, and therefore how accurately the target location distribution represents the uncertainty in target location.

In the following sections we provide mathematical definitions of the measures described above. The notions discussed above must be modified somewhat for the discrete distributions computed by Nodestar. In the discussion below we consider only the two-

dimensional marginal distribution on position. Fix a time t. Let X_t be the true target position at time t and p_{ij} be the probability computed by the Nodestar tracker that the target is in the ij^{th} cell of the two-dimensional marginal distribution on position.

2.2 AOU Size

For a Kalman filter the AOU size is equal to the area of the 2-σ ellipse of uncertainty about the mean estimated target position at time t. For Nodestar we find the AOU size by ordering the two-dimensional cell probabilities from highest to lowest. We add up these probabilities and the area in their cells until 86% is reached. The resulting cumulative area is the AOU size, the smallest region having 86% probability of containing the target.

To compare one tracker with another, we compute the average AOU size over all times at which the trackers produce estimated target locations.

2.3 Mean Missed Distance

For this measure we calculate the the root mean square (RMS) missed distance from the target's actual position at time t to the target distribution. The reason that we use RMS missed distance instead of mean missed distance is that the RMS missed distance is very easy to calculate for a bivariate normal distribution.

RMS Missed Distance for a Kalman Filter. Let Y_t be the mean position of the the bivariate normal distribution on target position produced by a Kalman filter at time t. Let σ_1 and σ_2 be the standard deviations of this distribution along its principal axes, and let

$$d = \text{distance from } X_t \text{ to } Y_t.$$

Then

$$\text{RMS missed distance} = \sqrt{d^2+\sigma_1^2+\sigma_2^2}\,. \tag{2.1}$$

RMS Missed Distance for Nodestar. For a Nodestar distribution we let

$$d_{ij} = \text{the distance from the target position } X_t \text{ to the center of the } ij^{th} \text{ cell.}$$

Then the mean squared missed distance, M_t^2, is computed as follows:

$$M_t^2 = \sum_{ij} p_{ij} d_{ij}^2. \tag{2.2}$$

and

$$\text{RMS missed distance} = \sqrt{M_t^2}. \tag{2.3}$$

For the comparison, we compute the average of the RMS missed distances for all times at which the trackers produce estimated target locations.

2.4 Accuracy

Before describing how we compute the accuracy of a statistical tracker, we first describe how containment statistics are computed.

Containment Statistic for a Kalman Filter. Let (x_1,x_2) and (y_1,y_2) be the coordinates of the positions X_t and Y_t in the coordinate system of the principal axes of the bivariate distribution that forms the estimate of target position at time t. Let σ_1 and σ_2 be the standard deviations along axes 1 and 2.

For a Kalman filter we determine the level curve (i.e., ellipse) of the bivariate normal position distribution on which X_t lies. Let $(z_1,z_2) = (x_1-y_1,x_2-y_2)$. This ellipse is characterized by the parameter k which satisfies the following relationship:

$$\frac{z_1^2}{\sigma_1^2} + \frac{z_2^2}{\sigma_2^2} = k. \tag{2.4}$$

The value of q_t, the containment statistic, is given by

$$q_t = 1 - \exp(-\frac{k^2}{2}). \tag{2.5}$$

Containment Statistic for Nodestar. For Nodestar, we compute the containment statistic q_t in the following manner. Let ij* be the cell containing the target at time t. Let

$$\hat{S} = \{ij \mid p_{ij} > p_{ij^*}\},$$

and let r be a random draw from a uniform [0, 1] distribution. The containment statistic, q_t, is computed as follows:

$$q_t = \sum_{ij \in S} p_{ij} + r\, p_{ij^*}. \tag{2.6}$$

Kolmogorov-Smirnov (K-S) Statistic. Having computed q_t from (2.5) or (2.6) for $t = 1\ldots,T$, we compute the K-S statistic as follows. Order the set $\{q_t\,;\, t = 1\ldots,T\}$ into $\{Q_1,\ldots,Q_T\}$ so that Q_1 is the smallest q_t, Q_2 the next, etc. Define the empirical distribution function F as follows:

$$\begin{aligned} F(x) &= 0 && \text{for } 0 \le x \le Q_1, \\ &= \frac{s}{T} && \text{for } Q_s < x \le Q_{s+1} \text{ for } 1 \le s \le T-1, \\ &= 1 && \text{for } Q_T < x \le 1. \end{aligned} \tag{2.7}$$

The K-S statistic D is given by

$$D = \max_{0 \le x \le 1} |F(x) - x|. \tag{2.8}$$

The statistic, D, measures the maximum deviation of the empirical distribution function F from the theoretical distribution function G defined by $G(x) = x$ for $0 \le x \le 1$, i.e., a uniform distribution on [0,1]. The distribution of D has been computed and tabulated (see reference [d]). The statistic D, called the K-S, statistic, can be used to test the hypothesis that q_t is uniformly distributed on [0,1]. This is not how we have used D. Instead we have used D to define accuracy in the manner described below.

Definition of Accuracy. The accuracy, A, is defined as

$$A = 100\,(1 - D)\%.$$

The statistic D has values that fall between 0 and 1. If D is close to 0, then there is very little deviation of the empirical distribution F from the theoretical one G. In this case the tracker is accurately representing the uncertainty in the estimate of the target's position and A will be close to 100%. If D is close to 1, then there is a large deviation of F from G and the accuracy is close to 0%.

Our objective is to use A to measure the relative accuracy of one tracker compared to another. If we applied the K-S test using D we would find that most trackers fail the test. This

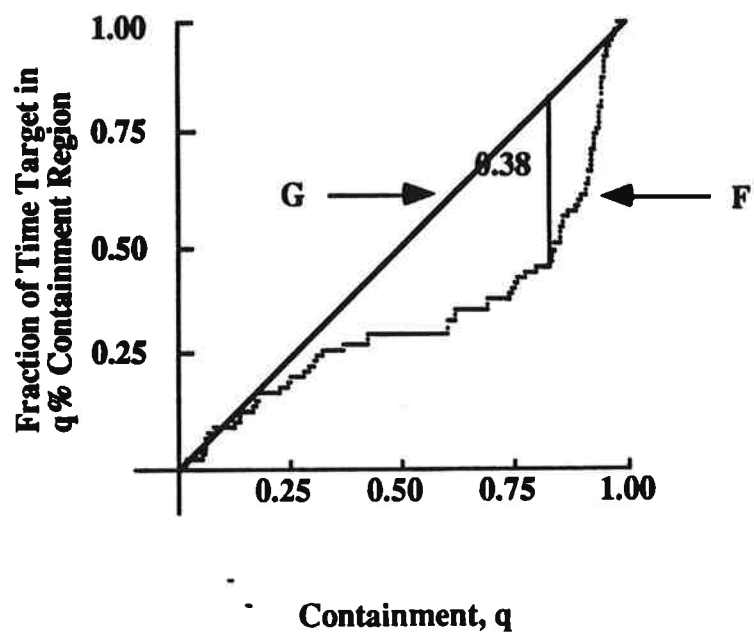
would not tell us much about the relative accuracy of one tracker compared to another. (Note, Nodestar passed the K-S test when used with the exercise data given in reference [c].) This would simply mean that because of the simplifications and idealizations that are necessary to build any tracker, the empirical function will eventually deviate from the theoretical one.

By viewing the plot of F compared to G (see Figure 1 for a sample plot), we can obtain further information about the quality of the estimates. If for example the empirical distribution F is consistently below G, then our probability maps are optimistic in the sense that they indicate better target localization than we are actually obtaining (see Figure 1). This would indicate that one of the modeling assumptions is too optimistic. Perhaps we are crediting one of the sensors with a smaller measurement error than it is actually experiencing. Similarly if F is consistently above G, then the tracker is being too pessimistic by presenting uncertainty areas that are too large.

The horizontal axis in Figure 1 is the value of the containment variable, q. The value of the vertical axis is the fraction of time the target is inside the q% containment region. Ideally, the target will be in the q% containment region q% of the time. If this were true, then we would have F=G and F would lie on the straight line representing G in Figure 1. In Figure 1, the K-S statistic is 0.38, and the accuracy is 62%.

Figure 1

SAMPLE K-S PLOT



CHAPTER III

TEST RESULTS

We designed a series of tests to compare the performance of MTST to Nodestar in the situation where there are multiple stationary arrays providing line-of-bearing detections on a single moving target. Thirty cases were examined. In each case a simulation was run to produce a series of detections on a specified target track. When a simulated detection occurred, the bearing to the target from the sensor was calculated. An error term was drawn from a normal distribution and added to the actual bearing to produce a "noisy" bearing which was recorded on a file. The noisy bearings simulate actual bearing calls made by the sensor. For each of the 30 cases, this file of bearings was fed into Nodestar and into MTST.

Both Nodestar and MTST require the user to make probabilistic assumptions about the target's motion. Since Nodestar uses detection / no-detection information from the arrays to help estimate the target's location, it requires an estimate of the target's source level at its primary (i.e., most detectable) frequency and the propagation loss curve for that frequency. It also needs estimates of all the remaining terms of the passive sonar equation in order to compute the detection / no detection likelihood functions. As well as comparing the performance of Nodestar to MTST, we also investigated the robustness of the trackers' performance when there is a mismatch between the assumptions used by the tracker (i.e., the motion and acoustic assumptions) and those used in the simulation that generated the file of detections fed into the trackers.

3.1 Scenario Description

In this section we describe the scenarios used in our test.

Simulator Motion Assumptions. For the simulator, we used the three base tracks shown in Figures 2 - 4. The figures also show the locations of the three fixed array sensors used to generate detections in the simulation.

Patroller. For the 10 patroller cases we used the base track shown in Figure 2. The speeds for each of the legs are given in Table 1. For each case we perturbed the length of each leg by making a random draw to adjust the length of the leg.

Constant Course and Speed. For the 10 constant-course-and-speed cases we used the target track shown in Figure 3 with speed 12 kts.

Evader. The base track for the evader motion is shown in Figure 4. The speeds for each of the legs are given in Table 2. For each case we perturbed the length of each leg by making a random draw to adjust the length of the leg.

Table 1

PATROLLER TRACKS

Leg	Course (deg)	Speed (kts)	Time on Leg (hrs) *
1	45	12	RAND(2.5)
2	270	11	RAND(2.0)
3	0	11	RAND(4.0)
4	90	10	RAND(3.0)
5	225	12	RAND(1.0)
6	135	12	RAND(1.5)
7	225	12	RAND(3.0)
8	280	10	RAND(5.5)

*Note, RAND(x) indicates a random draw from a uniform distribution over [x/2, 3x/2]. The base case has time on leg equal to x hrs.

Figure 2

PATROLLER BASE TRACK

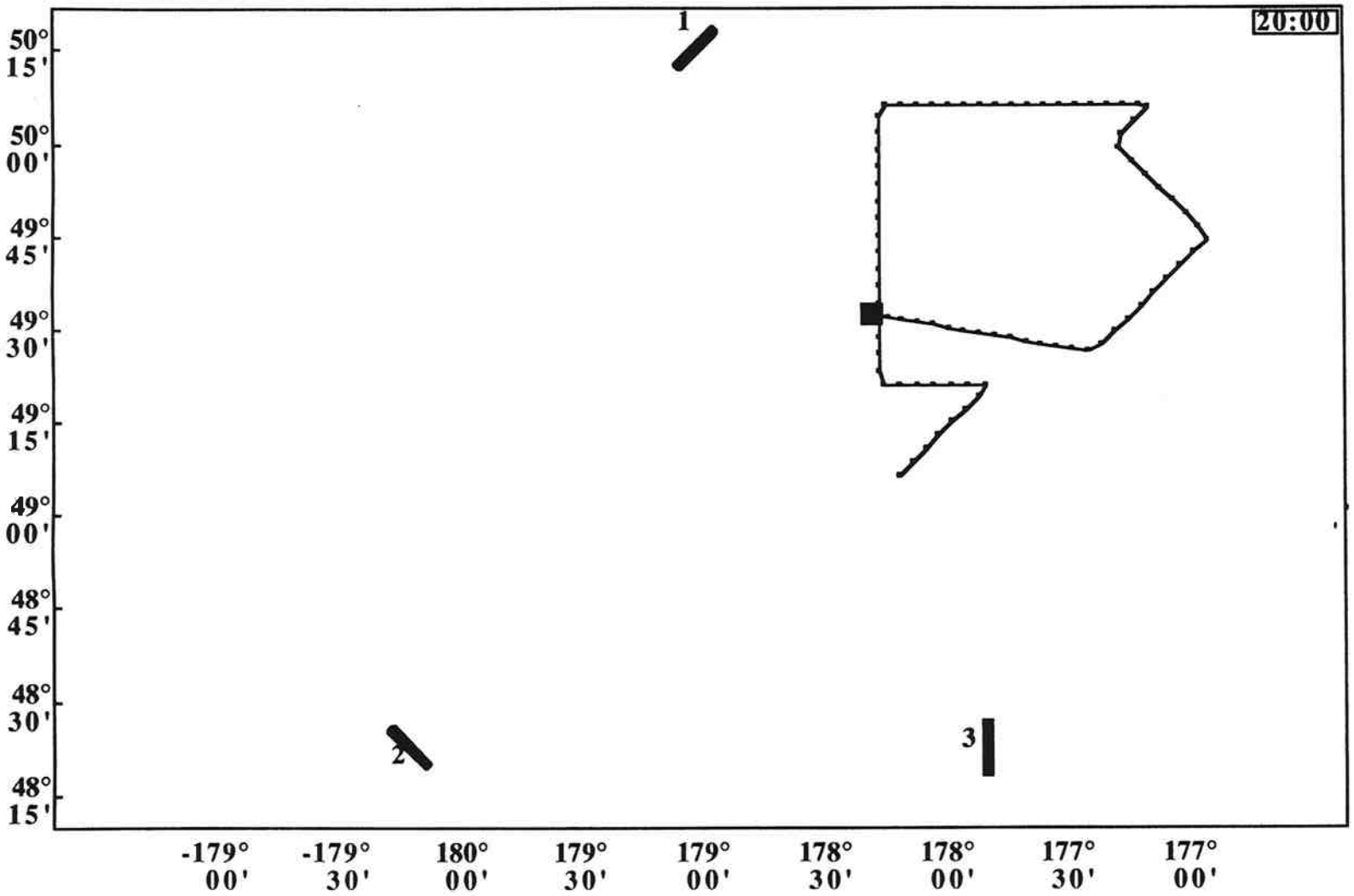


Figure 3

CONSTANT COURSE AND SPEED BASE TRACK

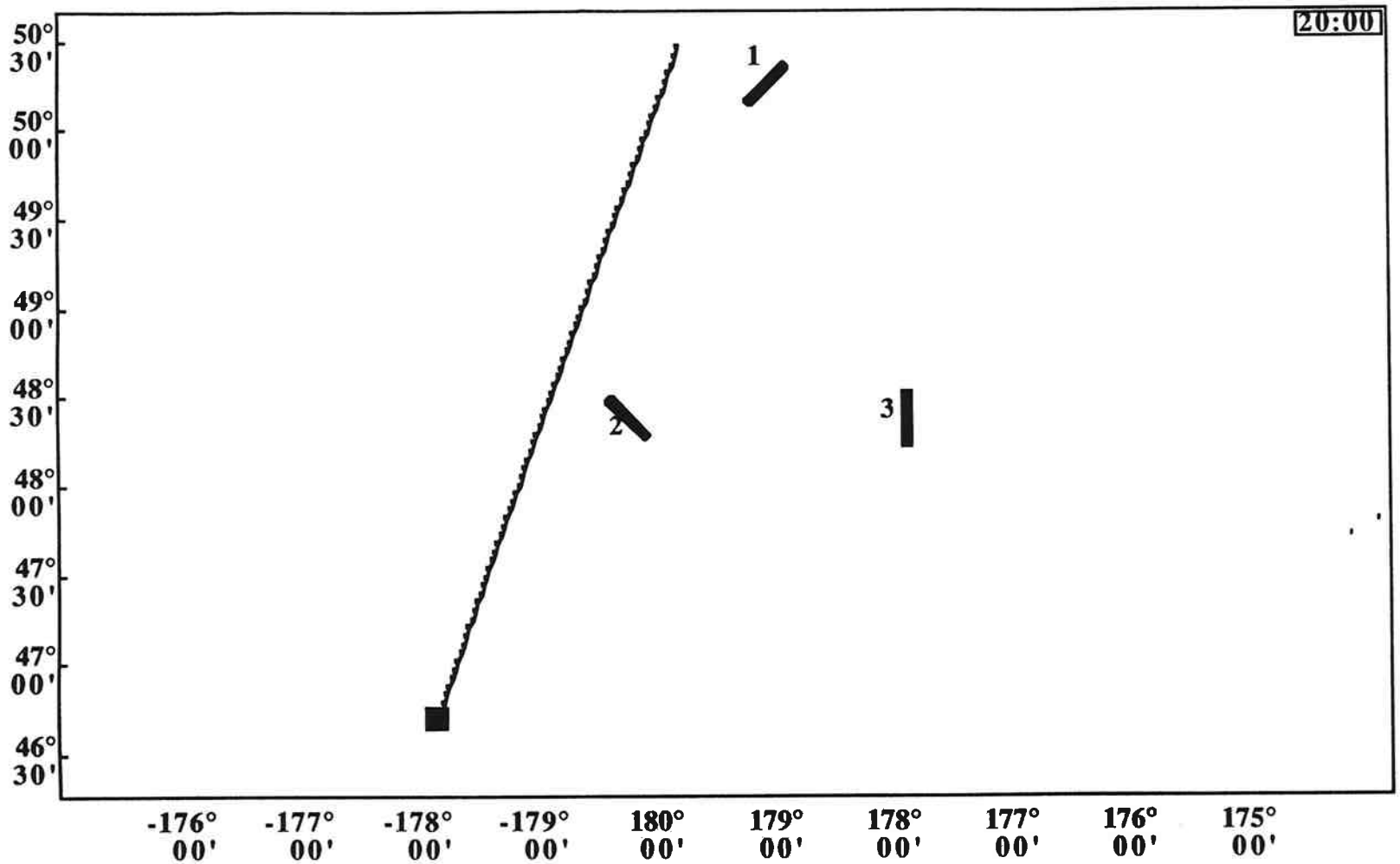


Figure 4

EVADING TARGET BASE TRACK

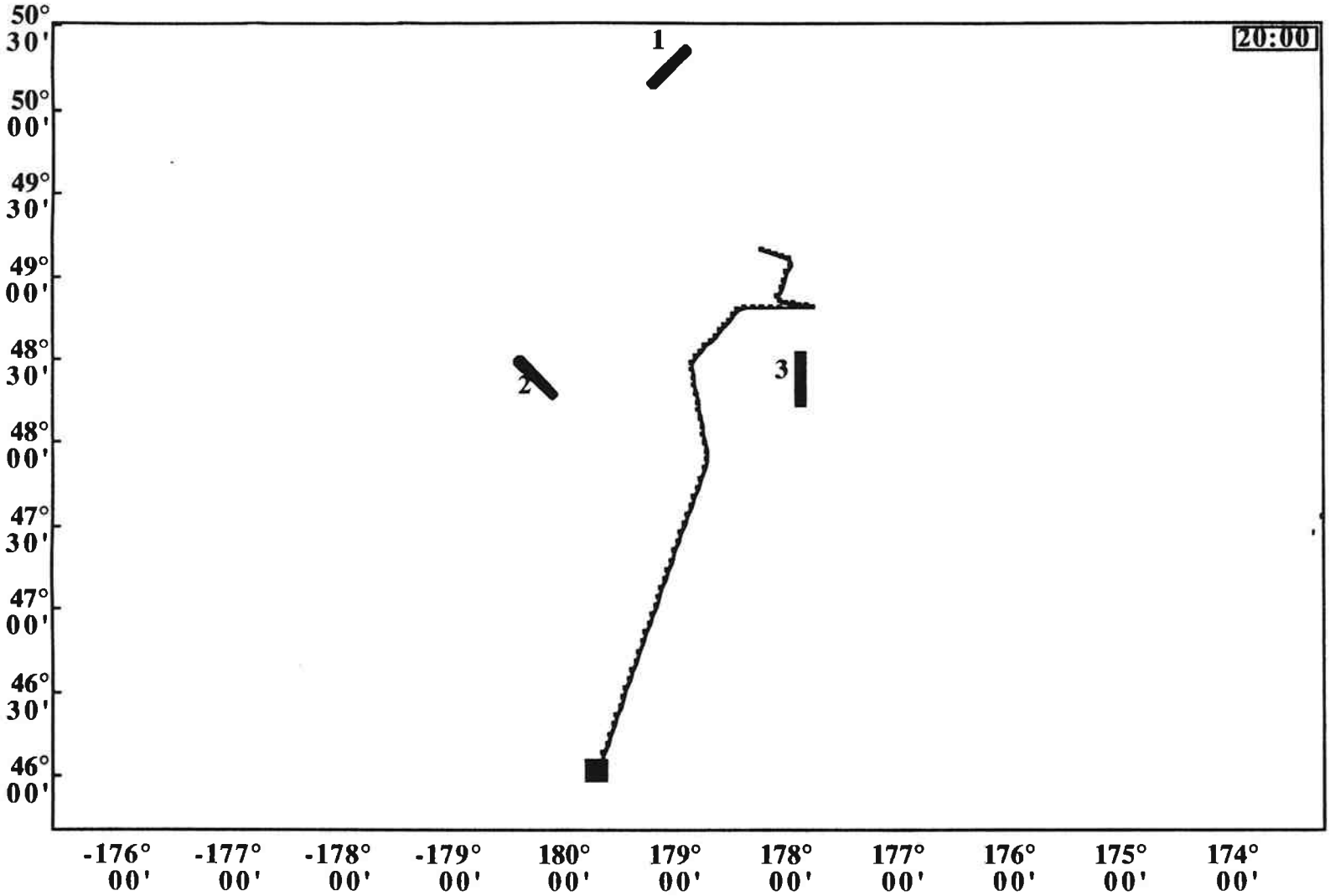


Table 2

EVADER TRACKS

Leg	Course (deg)	Speed (kts)	Time on Leg (hrs) *
1	110	10	RAND(1.0)
2	200	12	RAND(1.0)
3	100	10	RAND(1.0)
4	270	13	RAND(1.5)
5	220	10	RAND(2.0)
6	170	12	RAND(2.5)
7	200	14	RAND(12.0)

*Note, RAND(x) indicates a random draw from a uniform distribution over [x/2, 3x/2]. The base case has time on leg equal to x hrs.

Tracker Motion Assumptions. Both Nodestar and MTST used one fixed motion model for all 30 test cases.

Nodestar Motion Model. The Nodestar motion model assumed that the target's heading was uniform [0, 360] and its speed distribution uniform over [5kts, 20kts]. The target course changes were modeled as taking place at exponentially distributed times with mean time 1 hour between course changes.

MTST Motion Model. The motion model in MTST is an Integrated Ornstein-Uhlenbeck process. This process is specified by the relaxation parameter $\beta = 1/$ hr. and white noise scaling parameter $\sigma^2 = 156.25$ $(\text{nm})^2/\text{hr}^3$. The value of $\beta = 1/$ hr. corresponds to the assumption in Nodestar that the mean time between course changes is 1 hour. The value of σ was obtained by setting $\sigma/\beta^{1/2} = 12.5$ kts, the mean of the target speed distribution used by Nodestar. The choice of values for β and σ was made in conformance with the recommendations in reference [a]. The objective was to match the motion assumptions used by Nodestar as closely as possible.

The stochastic differential equation for an Integrated Ornstein-Uhlenbeck process is

$$dx(t) = v(t)dt$$

$$dv(t) = -\beta\, v(t) + \sigma\, \xi_t$$

where x(t) is the two-dimensional position at time t, v(t) is the velocity, and $\{\xi_t\,;\, t \geq 0\}$ is a white noise process.

Acoustic Environment. As well as varying the target motion, the test also considered the three acoustic environments represented by the propagation loss curves shown in Figures 5 - 7. The first is a surface duct environment, the second is direct path and bottom bounce, and the third a convergence zone environment. These curves are taken from reference [e]. The Figure-of-Merit (FOM) used by the simulator is shown on each of the propagation loss curves. Detections were simulated using a λ-σ model for acoustic fluctuations. (See reference [f].)

Mismatch of Tracker and Simulator Assumptions. The first type of mismatch that we consider is with the motion assumptions. Both Nodestar and MTST make assumptions about the target motion. We have chosen to have both Nodestar and MTST make the same motion assumptions (to the extent that we can match them) and hold them fixed throughout the 30 tests. The simulator motion assumptions vary widely from those used by the tracker. This variation tests the robustness of both trackers to mismatches between their motion assumptions and the target's actual motion.

The second type of mismatch concerns acoustic assumptions. MTST does not make any assumptions about the acoustical characteristics of the environment or the target. Nodestar requires that the user specify a propagation loss curve and an FOM for the target. To test the robustness of Nodestar to mismatches in the propagation loss curve and the FOM, we considered the following cases:

1. Correct transmission loss curve, correct estimate of figure of merit.
2. Correct transmission loss curve, high estimate of figure of merit.
3. Correct transmission loss curve, low estimate of figure of merit.
4. Incorrect transmission loss curve, correct estimate of figure of merit.

Table 3 gives a summary of the acoustic inputs.

Figure 5

SURFACE DUCT ENVIRONMENT

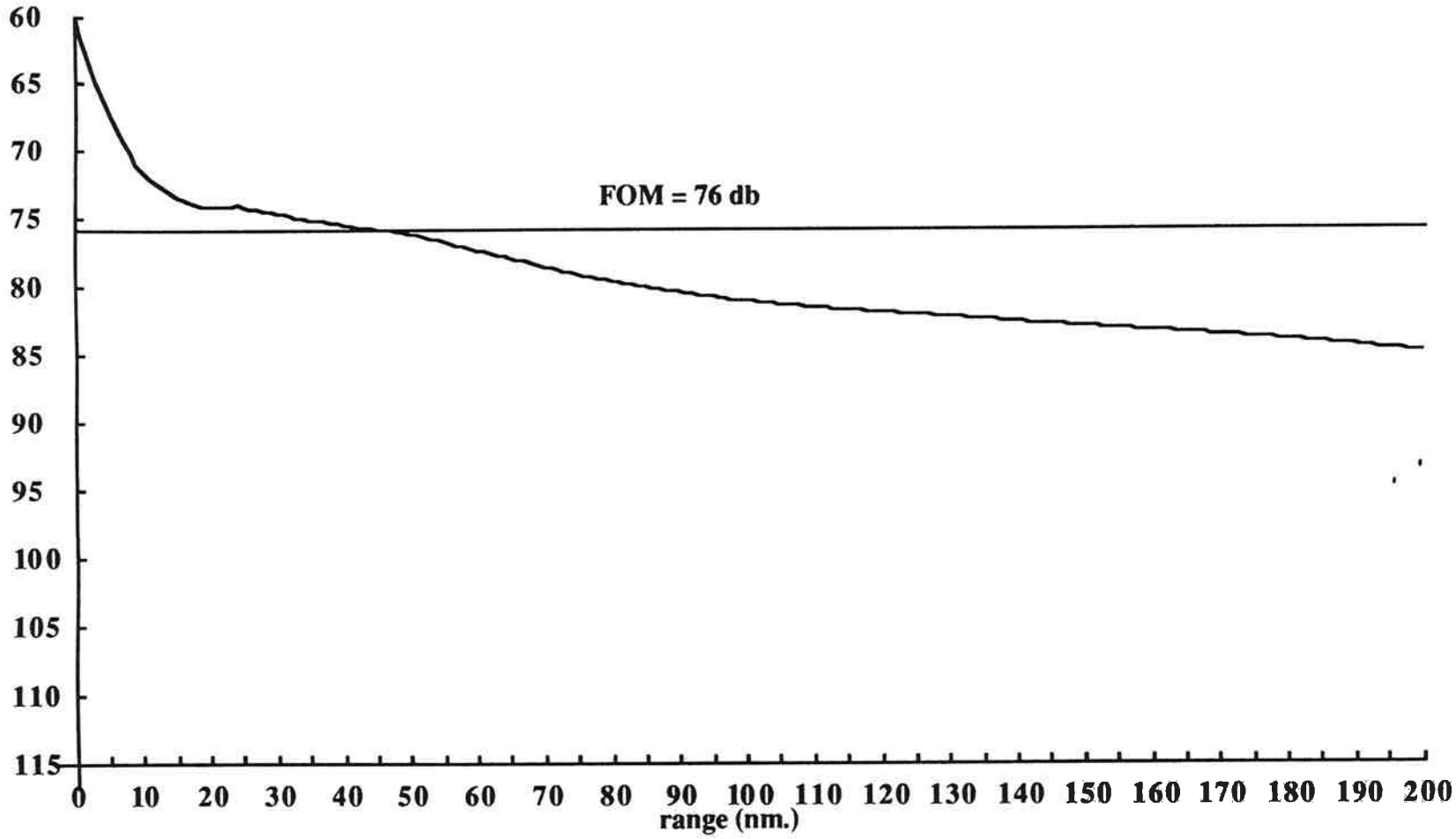


Figure 6

DIRECT PATH AND BOTTOM BOUNCE ENVIRONMENT

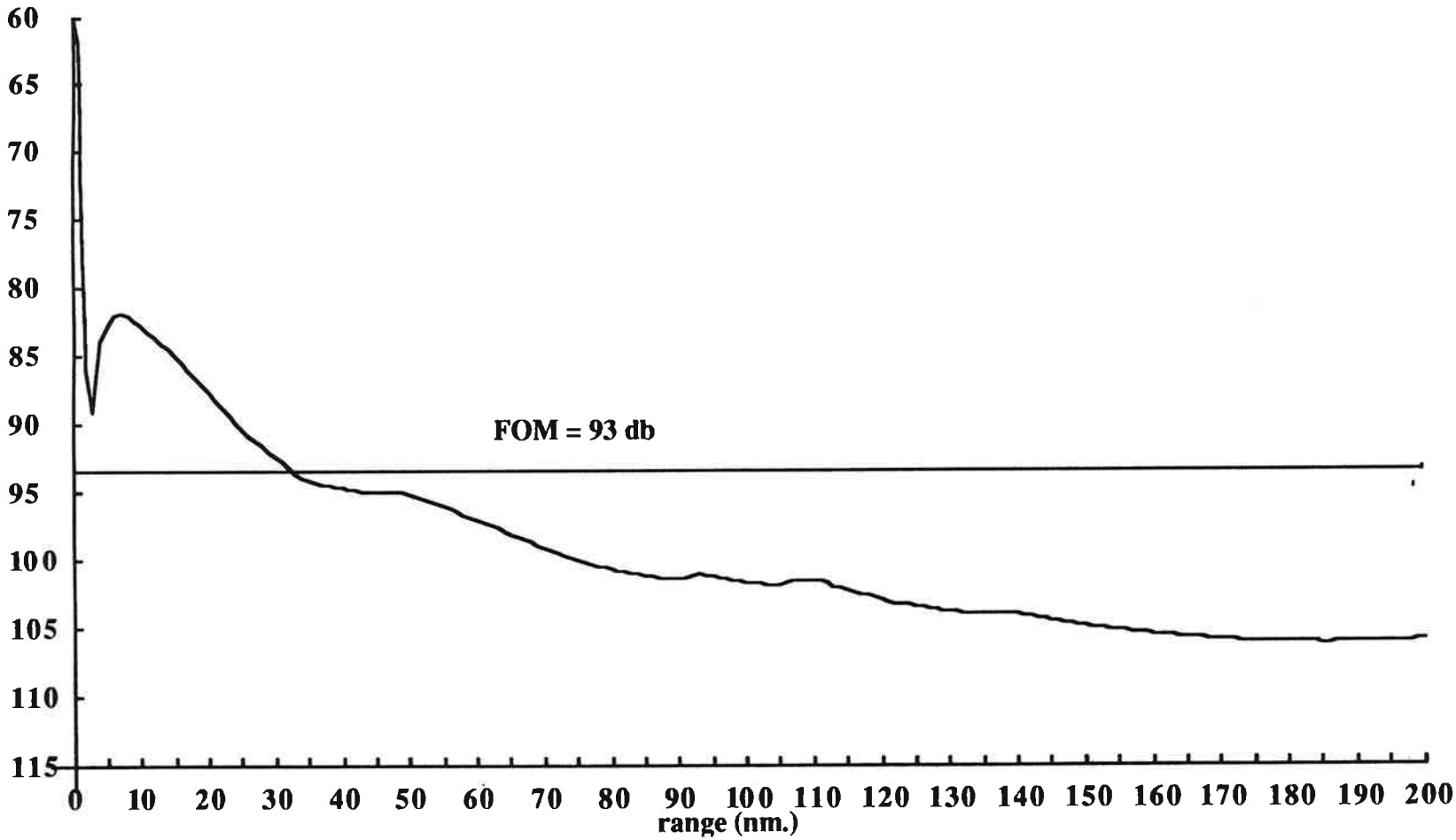


Figure 7

CONVERGENCE ZONE ENVIRONMENT

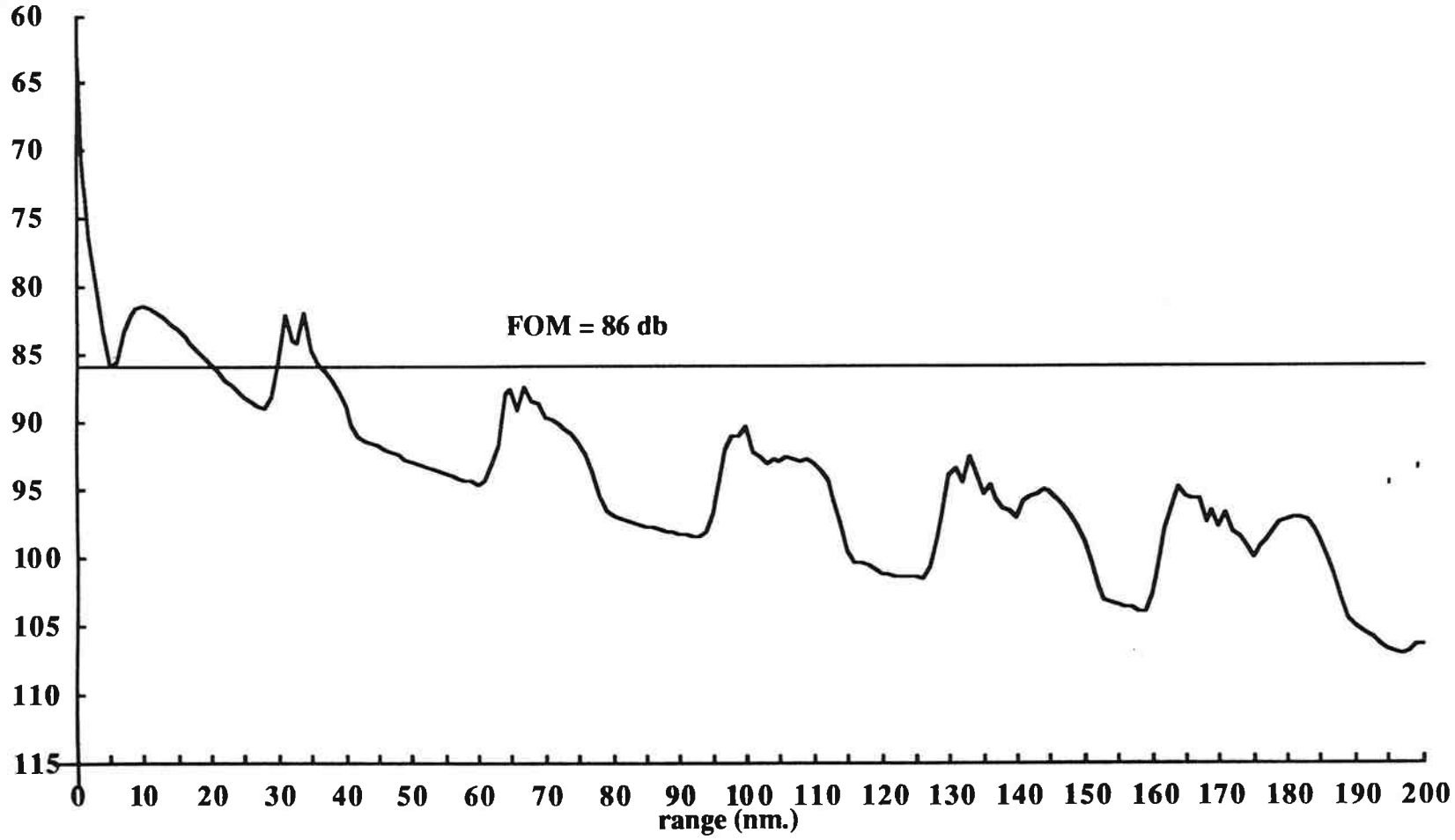


Table 3

SUMMARY OF ACOUSTIC INPUTS

Figure of Merit (db):	
In the Convergence Zone Environment	86.0
In the Direct Path/Bottom Bounce Environment	93.0
In the Surface Duct Environment	76.0
Standard Deviation of Acoustic Fluctuations (db)	3.5
High Estimate of Figure of Merit (db):	Correct FOM + 3.5
Low Estimate of Figure of Merit (db):	Correct FOM - 3.5
Rate of Acoustic Fluctuations (per hr.):	1.0
Mismatched Transmission Loss Curve:	
Simulator	Used convergence zone
Nodestar	Assumed direct path

3.2 Summary of Runs and Results

We ran one test for each combination of assumptions listed above. This produced 3 motion models × 3 propagation loss curves × 3 FOM assumptions + 3 mismatched propagation loss curves = 30 runs.

Each run was given a 3-letter label that describes it. Table 4 describes the meaning of the 3-letter label. The first letter identifies the acoustic environment; C stands for convergence zone, S for surface duct, and D for direct path. The second letter identifies the motion of the simulated target; P stands for patroller, E stands for evader, and C stands for constant course and speed. The third letter identifies the accuracy of the acoustic information; C stands for correct information, H stands for a high estimate of the FOM, L stands for a low estimate of the FOM, and I stands for an incorrect transmission loss assumption by Nodestar.

Table 4

RUN LABELS

First Letter Environment	Second Letter Target Motion	Third Letter Mismatch
C = CZ	P = Patroller	C = Correct Acoustic Assumptions
D = Direct Path, BB	C = Constant Course and Speed	H = High estimate of FOM
S = Surface Duct	E = Evader	L = Low estimate of FOM
		I = Incorrect Prop Loss

As an example, the run that involved a convergence zone environment, a target moving at a constant course and speed, and a high estimate of the figure of merit is given the label CCH.

Table 5 gives a summary of the results of the 30 runs. Each line of the table shows the values of the three MOE's (AOU size, mean missed distance (MMD), and accuracy) that were computed for Nodestar and MTST for the indicated run. Each run covered a period of 20 hours. Every 15 minutes each tracker produced an estimate of the target's location for a total of 81 times at which estimates were produced (including the initial estimate at time 0). The better of the two numbers in each category for a run is labeled with an asterisk. A quick glance shows that Nodestar wins most of the comparisons. The columns labeled G and H give the number of times at which contact was gained and held respectively during a run. The sum of the gains and holds yields the total number of contacts made by all three arrays during the run.

The first row of Table 5 shows the results for the case of a convergence zone environment, constant-course-and-speed target, and correct estimates by the Nodestar of the FOM and propagation loss curve. Note that Nodestar has a lower average AOU size, smaller mean missed distance (MMD), and higher accuracy than MTST. The next two lines show the results for two cases with the same target motion and environmental characteristics as the first one, but with Nodestar assuming that the FOM is one standard deviation higher and lower respectively than it actually is in the simulation. *Even with this mismatch in acoustic assumptions, Nodestar wins in all three categories.*

In the third line, run CCL, the AOU size and MMD for Nodestar and MTST are smaller than in the first run (CCC) where Nodestar was using the correct acoustic assumptions. The reason for this may be found in the number of gains and holds. (Each run uses a different random number seed and a slightly different target track.) In the CCL run there were 10 gains and 49 holds for a total of 59 detections. This compares with a total of 17 detections for the CCC case. The better MOEs for CCL reflect the fact that when the trackers receive more detection information, they perform better. Even though both trackers improve, Nodestar still outperforms MTST.

One would expect Nodestar to outperform a Kalman filter such as MTST in a convergence zone environment. The results in Table 5 bear that out. In 27 comparisons, there are only three instances in which MTST has a better MOE than Nodestar. These are the accuracy MOEs for runs CEH, CEL, CPC. For the CEH run, the accuracies of Nodestar and MTST are very close, *but Nodestar produces a substantially smaller AOU size and MMD*. In runs CEL and CPC, MTST has higher accuracy, but Nodestar provides the smaller AOU size and MMD. Overall, Nodestar is clearly superior.

In the case of a direct path or ducting environment one might expect that Nodestar's edge over MTST would diminish because the acoustic information in this environment would not be as valuable as in a convergence zone environment. *Surprisingly, the results in the table below show that Nodestar continues to outperform MTST in direct path and surface duct environments in the face of mis-estimates of FOM.*

Examining the three runs (DCI, DEI, and DPI) in which the propagation loss curve was convergence zone but Nodestar thought it was direct path, we see that Nodestar bettered MTST in 8 out of the 9 MOE comparisons.

The obvious conclusion is that Nodestar performs much better than MTST in favorable situations (e.g., convergence zone environments with correct acoustic estimates) and continues to outperform MTST in cases where Nodestar is given a poor estimate of FOM or even the wrong propagation loss curve. In the following section we test this conclusion statistically.

Figures A.1 - A.30 in Appendix A show the K-S graphs of Nodestar vs MTST for each of the 30 cases. For most of the cases Nodestar is clearly superior. In most cases the empirical distribution function lies below the line corresponding to $G(x) = x$. This indicates that both trackers tended to be optimistic in their estimated degrees of target localization.

Table 5

SUMMARY RESULTS

Run	AOU (nm^2)		MMD (nm)		Accuracy (%)			
	Nodestar	MTST	Nodestar	MTST	Nodestar	MTST	G	H
CCC	2,839*	4,880	36.83*	64.76	69.76*	27.62	4	13
CCH	2,354*	4,158	33.47*	54.91	79.91*	33.36	5	16
CCL	946*	1,122	23.41*	28.74	70.19*	45.18	10	49
CEC	1,571*	2,530	25.62*	33.39	67.19*	65.17	6	52
CEH	2,874*	5,822	41.61*	69.11	52.34	53.88*	7	8
CEL	2,908*	4,641	46.84*	46.94	46.38	59.28*	4	23
CPC	4,597*	7,584	41.07*	45.31	67.58	90.45*	3	9
CPH	4,017*	7,247	32.59*	49.24	88.3*	65.90	3	19
CPL	11,080*	16,860	56.83*	65.19	84.48*	81.9	2	0
DCC	1,057*	1,743	21.73*	48.21	50.7*	30.39	5	32
DCH	4,852*	8,849	66.21*	74.38	19.04	41.13*	2	7
DCL	1,911*	3,620	36.07*	58.60	25.61	27.99*	3	26
DCI	1,482*	2,595	21.45*	50.62	63.57*	36.42	6	25
DEC	3,705*	6,688	40.86*	63.76	71.65*	61.95	3	5
DEH	2,012*	3,697	24.07*	32.63	71.41	79.19*	4	29
DEL	3,768*	6,147	66.64*	96.15	30.05*	25.37	2	13
DEI	4,664*	8,129	66.68*	70.99	27.02	44.83*	4	5
DPC	898*	1,322	21.55*	28.90	70.83*	68.99	9	24
DPH	2,155*	3,831	36.98*	41.28	41.42	70.93*	6	16
DPI	2,670*	4,706	26.82*	39.80	72.15*	61.60	4	26
DPL	6,680*	10,940	53.20*	60.94	62.53*	58.07	2	0
SCC	5,668*	8,162	72.26	63.92*	39.96	54.45*	7	3
SCH	1,224*	1,529	22.34*	39.37	83.21*	40.36	8	30
SCL	5,755*	9,273	70.14*	75.94	53.64*	43.45	3	4
SEC	24,920	21,800*	181.50	133.10*	9.77	32.00*	0	0
SEH	3,275*	5,139	30.06*	65.81	89.13*	50.01	4	22
SEL	3,160*	4,410	41.55*	45.08	65.58*	65.25	5	16
SPC	1,750*	2,587	27.43*	34.01	71.12*	70.91	8	13
SPH	1,639*	2,310	29.66*	40.33	62.48*	50.30	6	20
SPL	860*	1,261	15.97*	33.90	89.47*	43.12	11	22

* indicates the winner.

3.3 Wilcoxen Test for Paired Observations

The Wilcoxen test is a classic and powerful non-parametric test used to compare observations that occur in pairs without having to know the distribution of the pairs of statistics. (See reference [g].) This test is often used to compare the performance of one medical treatment to another. In our case, we have identical detection files that were fed to two different trackers for 30 runs. For each run, three MOEs are computed. Each MOE represents a paired comparison of Nodestar to MTST. As we look at each of the MOEs, AOU size, MMD, and accuracy, it appears that Nodestar performs better than MTST over the 30 runs. The Wilcoxen, paired-observations test allows us to test this hypothesis statistically. Our null hypothesis is that both Nodestar and MTST perform equally well and the observation that Nodestar beats MTST so many times and by such margins is due strictly to chance.

Description of the Wilcoxen test. We assume that there are n test cases and that each tracker produces a single number for each measure of performance for each case. For the area of uncertainty, this number is the average area in the 86% containment region. For the mean missed distance, this number is the average root mean squared missed distance; and for the accuracy, this number is the accuracy measurement.

We consider the three measures of performance separately and assume that we have n pairs $\{(N_i, M_i) \mid i = 1,\ldots, n\}$, corresponding to one particular measure of performance. We form the set of differences $\{D_i = N_i - M_i \mid i = 1,\ldots,n\}$ and sort the D_i's by ascending absolute value. With respect to this sorting, let r(i) be the rank of D_i. For the comparisons of AOU size and MMD, we add up the ranks of the negative D_i's In other words we calculate

$$W = \sum_{D_i<0} r(i).$$

If there is no tendency for one tracker to have lower values of the measure of performance than the other, we would expect W to be equal to roughly half of the sum of the ranks from 1 to n, i.e. n(n+1)/4. If W is smaller than n(n+1)/4 , then we would have evidence that Nodestar produces MOEs (e.g., AOU size or MMD) that are smaller than those produced by MTST.

For values of $n \geq 30$, W is approximately normal with mean n(n+1)/4 and variance (n+1)(2n+1)/24. Let $C(\alpha,n)$ denote the critical value of the statistic W with a sample of size n

and a significance level of α. We can determine $C(\alpha,n)$ from the critical values of a normal distribution.

> If $W < C(\alpha, n)$, we reject the null hypothesis that there is no difference between trackers at the α% significance level.

To test the significance of the difference between Nodestar and MTST in accuracy, we simply take the negative of the accuracy MOEs and apply the test to them.

Results.

<u>AOU Size:</u> W = 24. We reject the hypothesis that Nodestar and MTST produce the same AOU size at the 1% significance level.

<u>MMD:</u> W = 43. We reject the hypothesis that Nodestar and MTST produce the same MMD at the 1% significance level.

<u>Accuracy:</u> W = 153. We reject the hypothesis that Nodestar and MTST produce the same accuracy at the 5% significance level.

The tests confirm what see qualitatively. Nodestar performs significantly better than MTST in AOU size, mean missed distance, and accuracy over the 30 runs.

3.4 Conclusions

These tests show that Nodestar performs better than MTST over a wide variety of situations. These include a variety of target motion behaviors and acoustic environments. In many cases Nodestar performed markedly better than MTST. In no case did MTST perform consistently better than Nodestar. The superiority of Nodestar remained true when it was using bad estimates of the FOM and even when it was using the wrong propagation loss curve.

The conclusion is that one never pays a penalty by using Nodestar and that in many cases one obtains substantially better performance from Nodestar than from a Kalman filter such as MTST.

One might ask whether this difference between Nodestar and MTST could be attributed to the possibility that MTST is not a good implementation of a Kalman filter. In our minds the answer to this question is clearly no. MTST is one of the best Kalman filters for submarine tracking. It has a motion model that provides sensible velocity distributions. It handles non-

linear information in a reasonable fashion given the linear constraints imposed by the Kalman methodology. In addition, MTST was subjected to numerous tests against competing Kalman algorithms before being selected for use in the successor to the Navy's Outlaw Shark system, the Shore Targeting Terminal.

When the output from Nodestar and MTST are viewed side by side at each time step, the reasons that MTST performs so poorly compared to Nodestar become clear. First, the lack of range information in the bearing detections used by MTST makes it difficult for MTST to localize the target. The linear approximations required to incorporate line-of-bearing detections causes the target distribution to get drawn into the sensor in a sort of fatal attraction. The inability to use negative information allows the target distribution to expand into regions near the sensor over long periods when no contact is obtained. All of these problems will be shared by any Kalman filter. They are not specific to MTST.

REFERENCES

[a] *Technical Documentation for the TIMS/MRS Implementation of the Maneuvering Target Statistical Tracker (MTST) Revision 7.0 (T/M).* Daniel H. Wagner Associates Memorandum to Naval Tactical Support Activity (NTSA) by N. L. Gerr, 31 January 1984.

[b] *Stochastic Processes and Filtering Theory* by Andrew H. Jazwinski, Academic Press, New York, 1970.

[c] *Sea SALT: Test and Description*, Metron Report to Naval Sea Systems Command (NAVSEA 63D) by L. D. Stone, 30 June 1989 (Case 6440-042).

[d] *Handbook of Tables for Probability and Statistics* by William H. Beyer, CRC Press, Boca Raton Florida, 1968 (Second Edition).

[e] *SEATAC Notional Acoustic Transmission Loss Functions*, SEATAC Memorandum #1-77 by James J. Galvin, LCdr., USN, Naval Warfare Analysis Group, Center for Naval Analyses, Arlington, VA.

[f] *Theory of Cumulative Detection Probability.* D. H. Wagner, Associates Report to Naval Underwater Sound Laboratory, by E. P. Loane, H. R. Richardson, and E. S. Boylan, November 10, 1964.

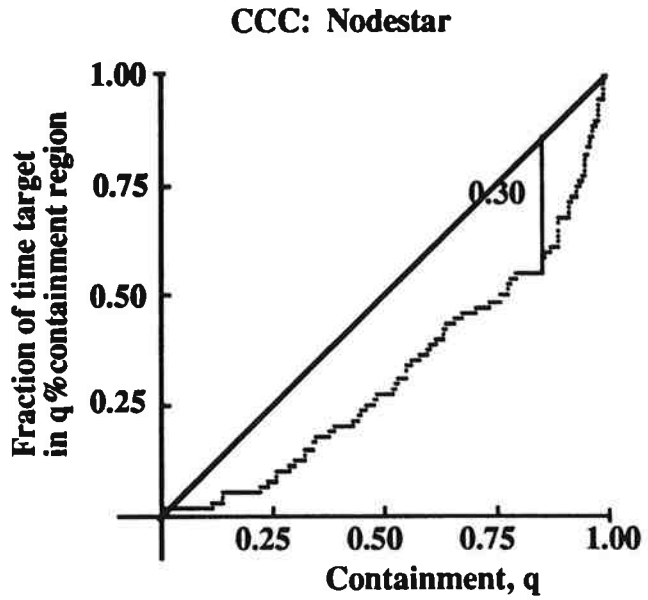
[g] *Nonparametric: Statistical Methods Based on Ranks* by E. L. Lehmann, Holden-Day, Inc., 1975.

APPENDIX A

CONTAINMENT STATISTIC PLOTS

This appendix presents the plots of the empirical distribution function for the containment statistic for Nodestar and MTST for the 30 runs summarized in Table 5.

Figure A.1


CCC: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.30
0.25
0.50
0.75
1.00
Containment, q

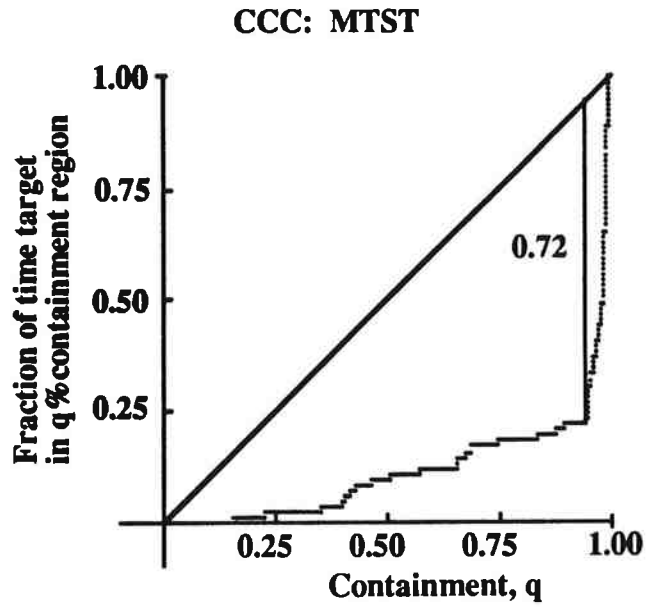
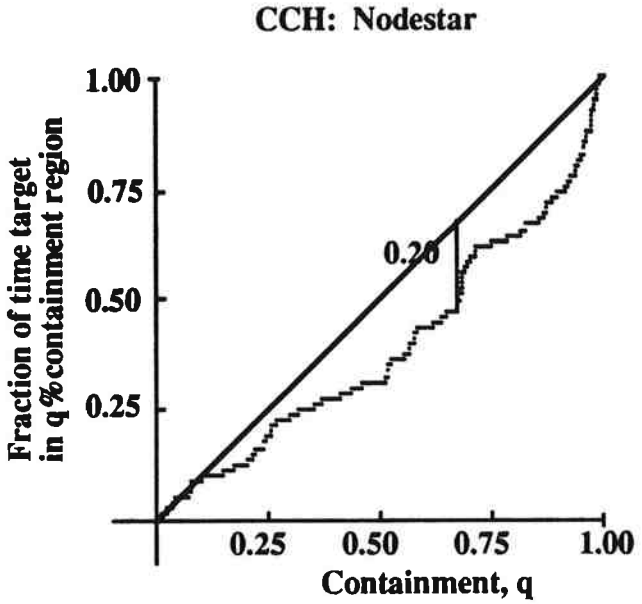

CCC: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.72
0.25
0.50
0.75
1.00
Containment, q

Figure A.2


CCH: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.20
0.25
0.50
0.75
1.00
Containment, q

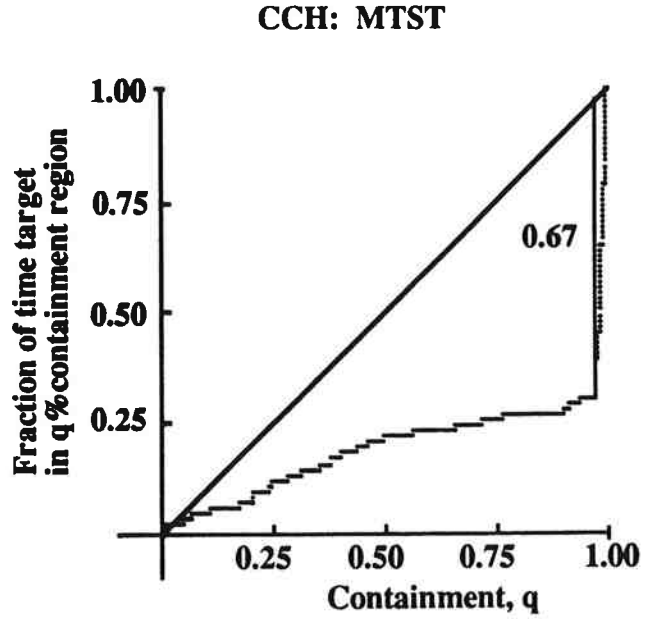
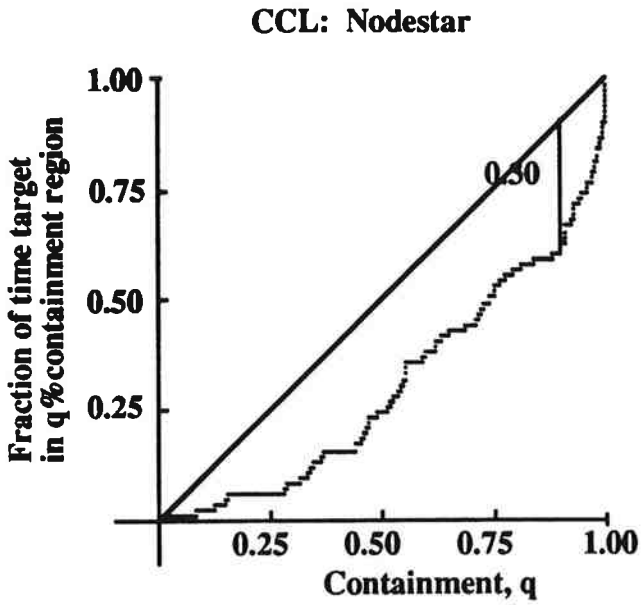

CCH: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.67
0.25
0.50
0.75
1.00
Containment, q

Figure A.3


CCL: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.50
0.25
0.50
0.75
1.00
Containment, q

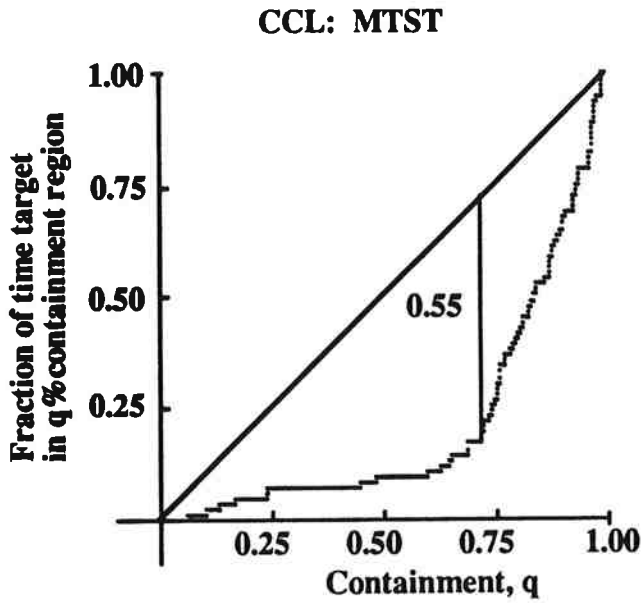
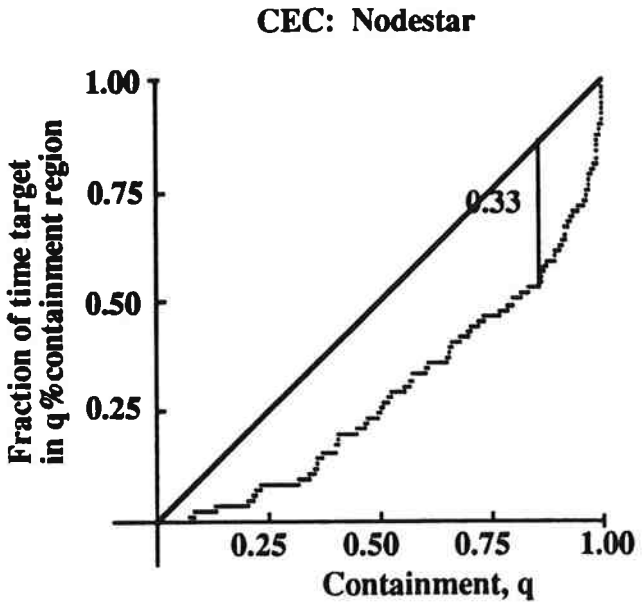

CCL: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.55
0.25
0.50
0.75
1.00
Containment, q

Figure A.4


CEC: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.33
0.25
0.50
0.75
1.00
Containment, q

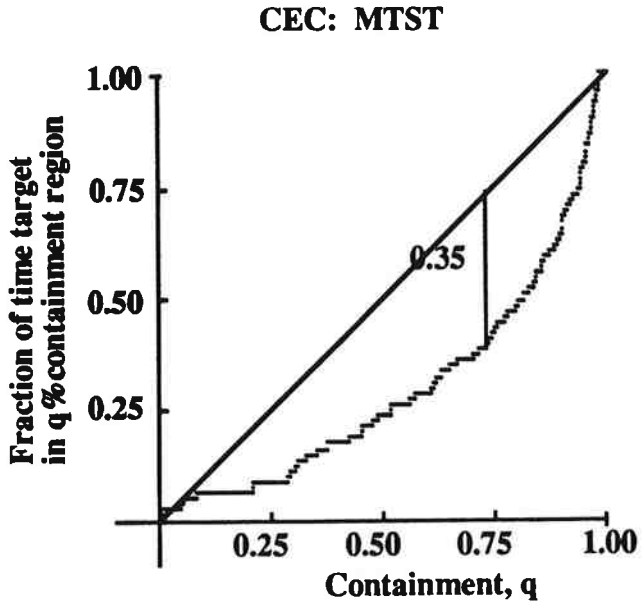
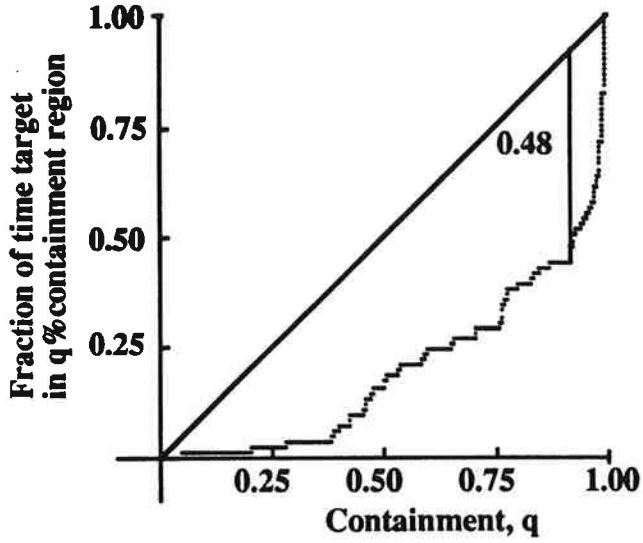

CEC: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.35
0.25
0.50
0.75
1.00
Containment, q

Figure A.5

CEH: Nodestar



CEH: MTST

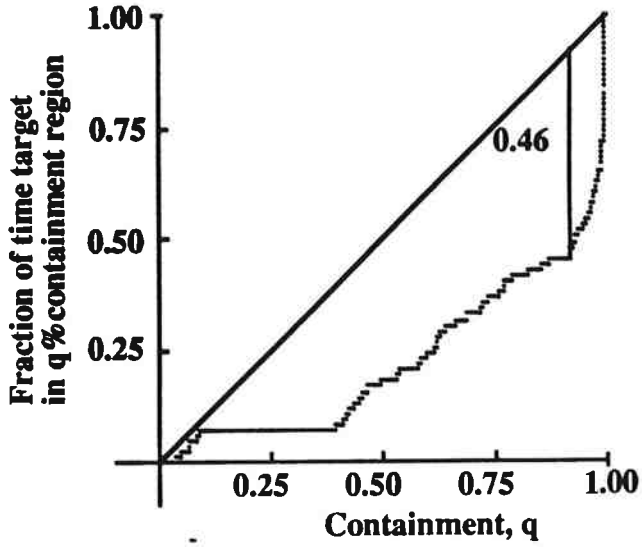
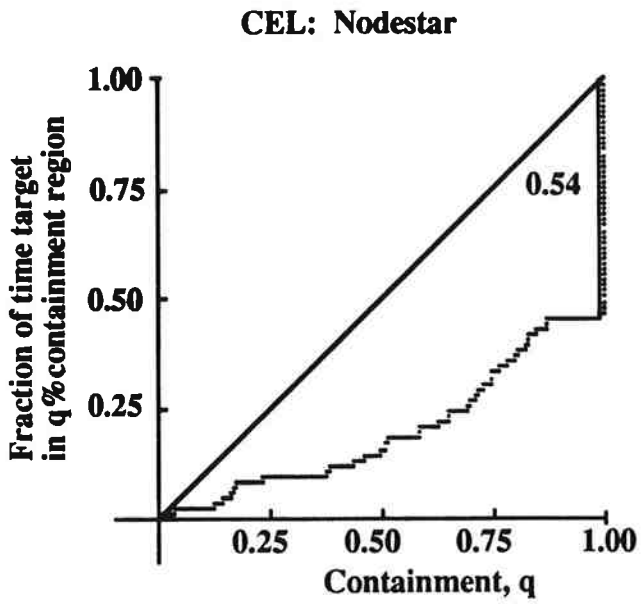


Figure A.6


CEL: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.54
0.25
0.50
0.75
1.00
Containment, q

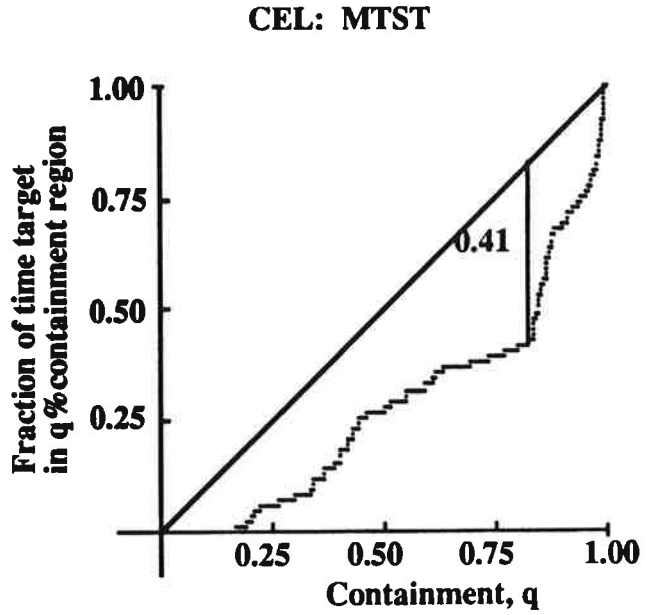
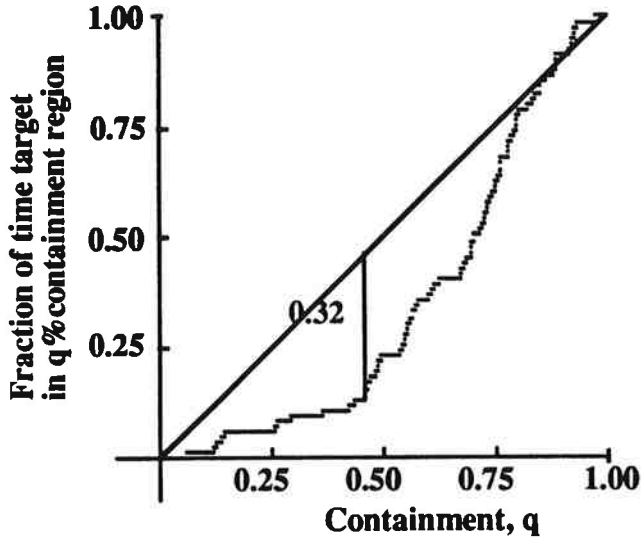

CEL: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.41
0.25
0.50
0.75
1.00
Containment, q

Figure A.7

CPC: Nodestar



CPC: MTST

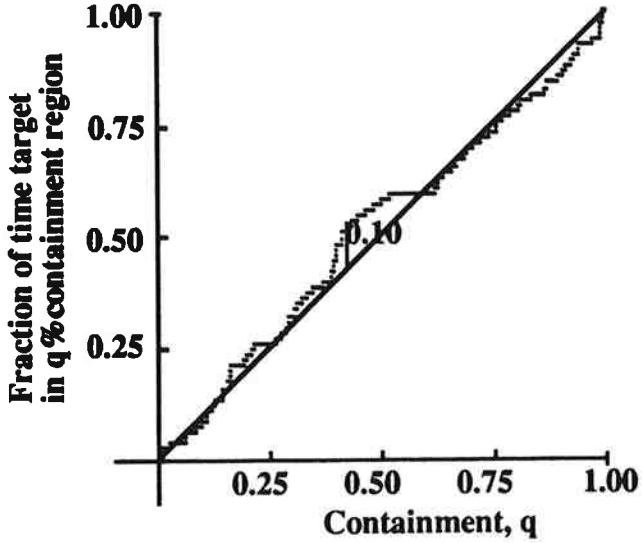
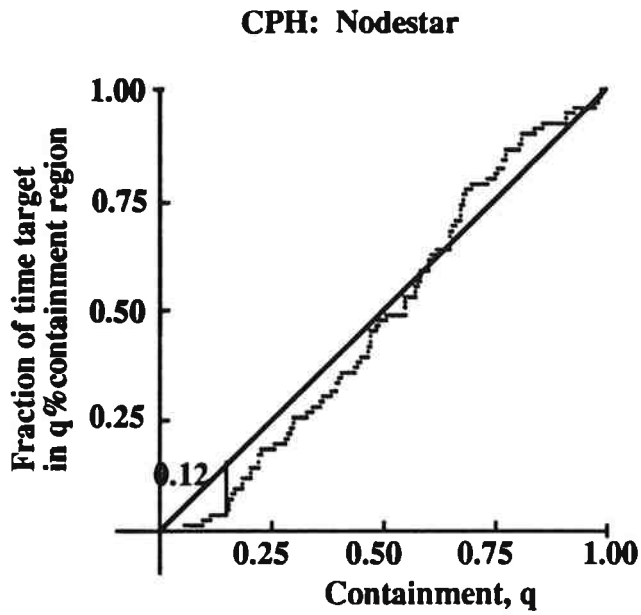


Figure A.8


CPH: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.12
0.25
0.50
0.75
1.00
Containment, q

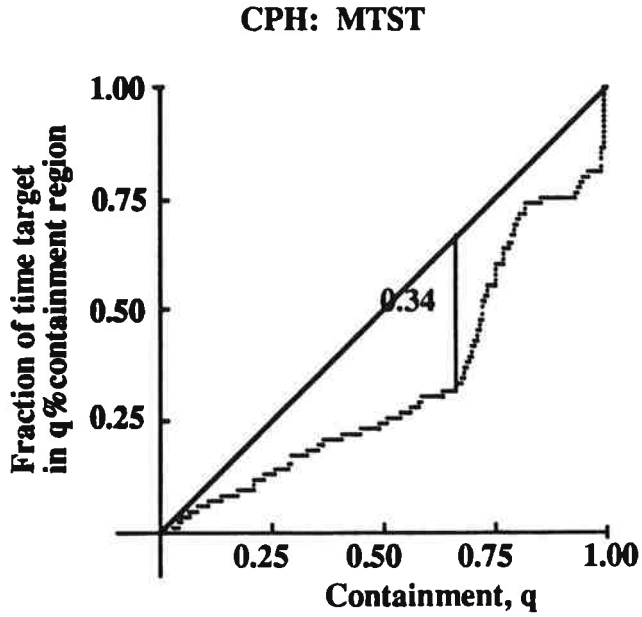
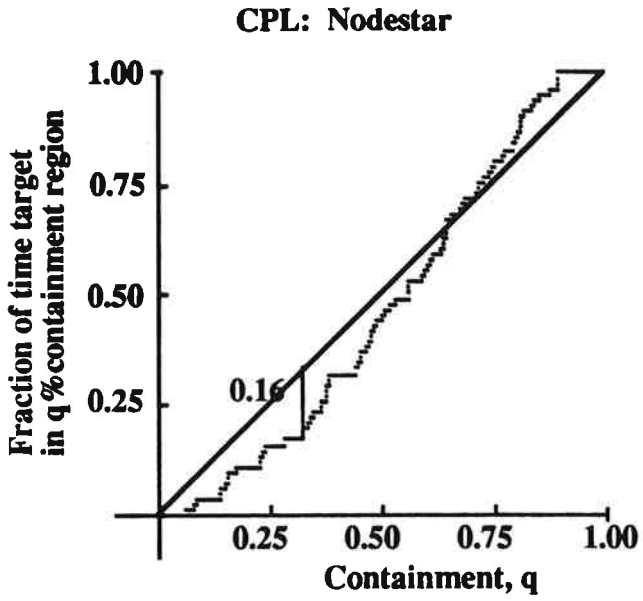

CPH: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.34
0.25
0.50
0.75
1.00
Containment, q

Figure A.9


CPL: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.16
0.25
0.50
0.75
1.00
Containment, q

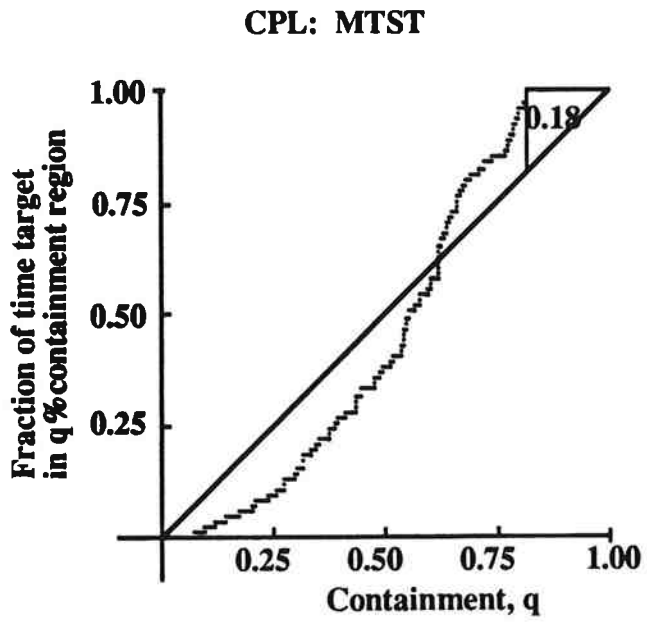
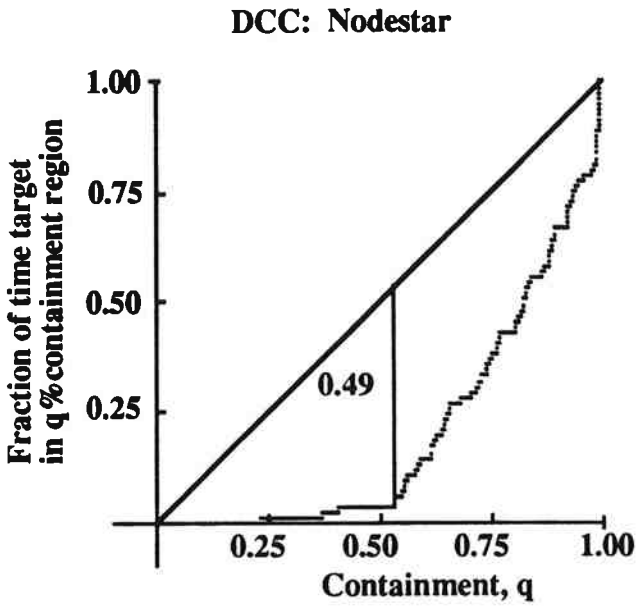

CPL: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.18
0.25
0.50
0.75
1.00
Containment, q

Figure A.10


DCC: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.49
0.25
0.50
0.75
1.00
Containment, q

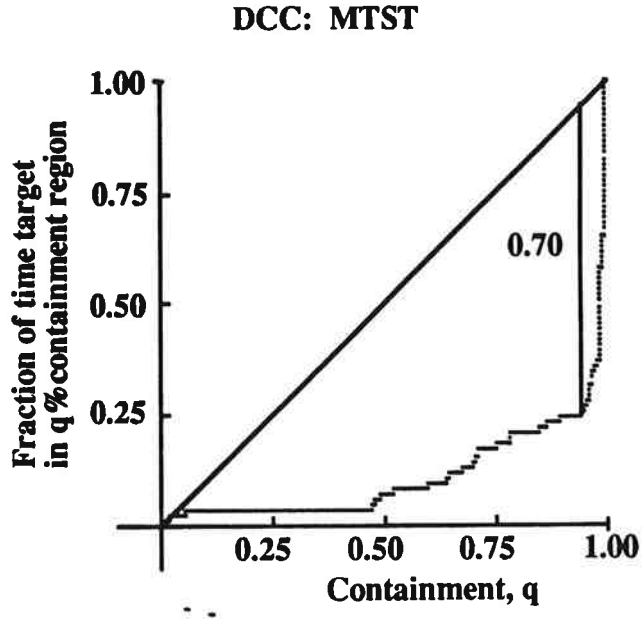
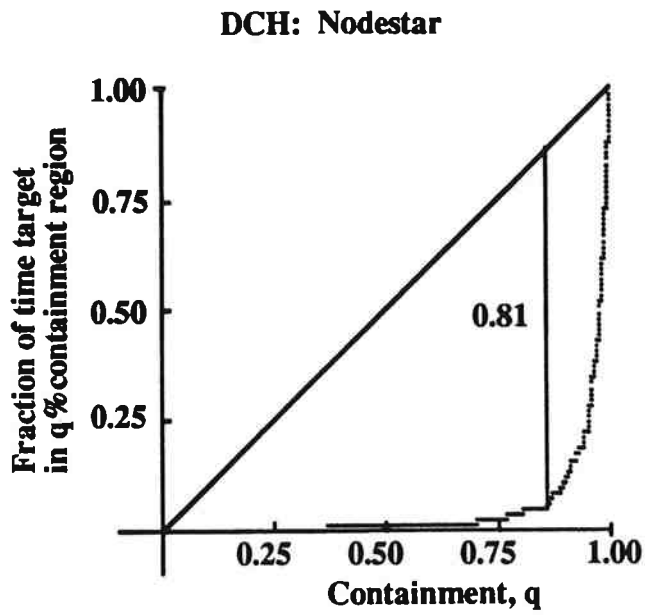

DCC: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.70
0.25
0.50
0.75
1.00
Containment, q

Figure A.11


DCH: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.81
0.25
0.50
0.75
1.00
Containment, q

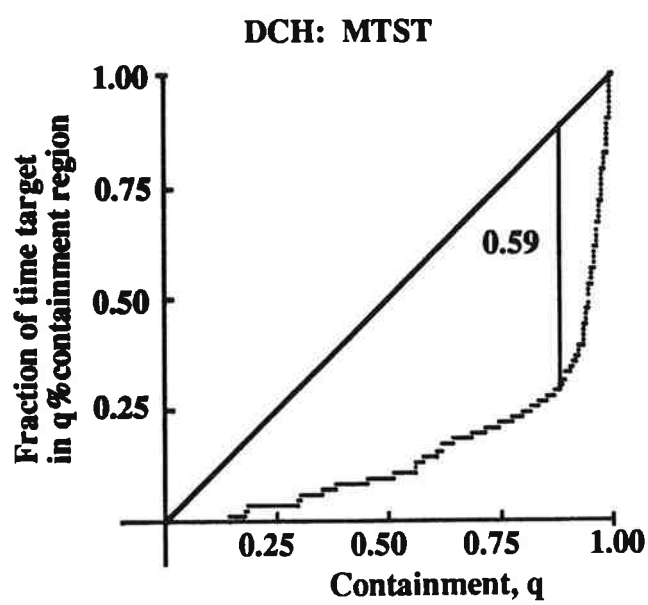
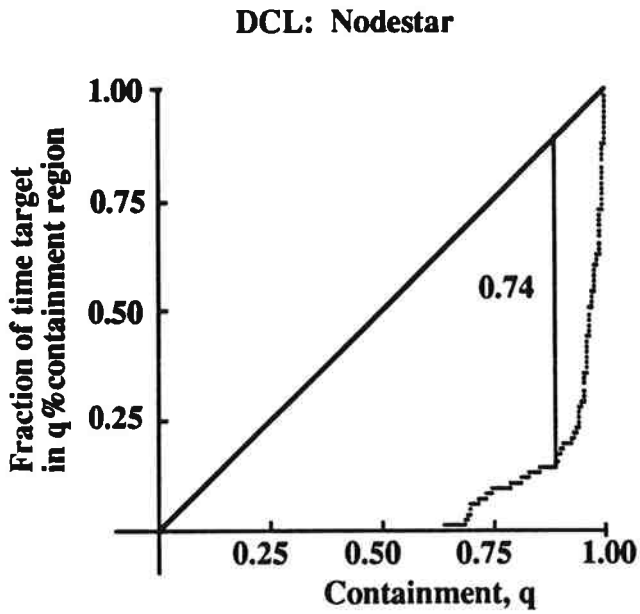

DCH: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.59
0.25
0.50
0.75
1.00
Containment, q

Figure A.12


DCL: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.74
0.25
0.50
0.75
1.00
Containment, q

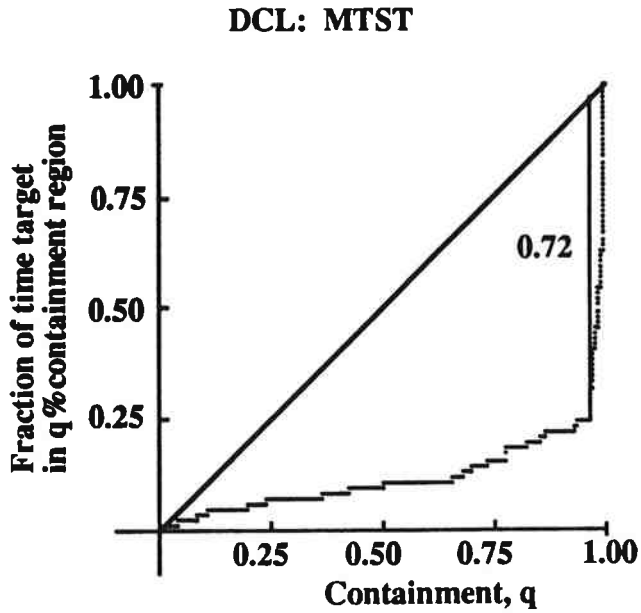
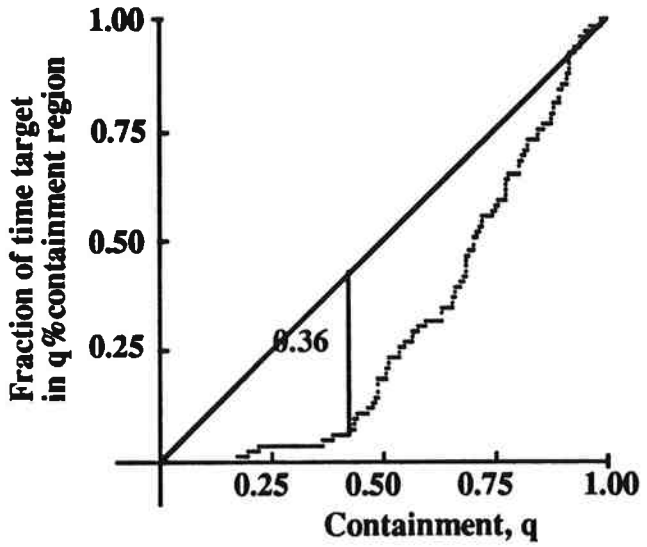

DCL: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.72
0.25
0.50
0.75
1.00
Containment, q

Figure A.13

DCI: Nodestar



DCI: MTST

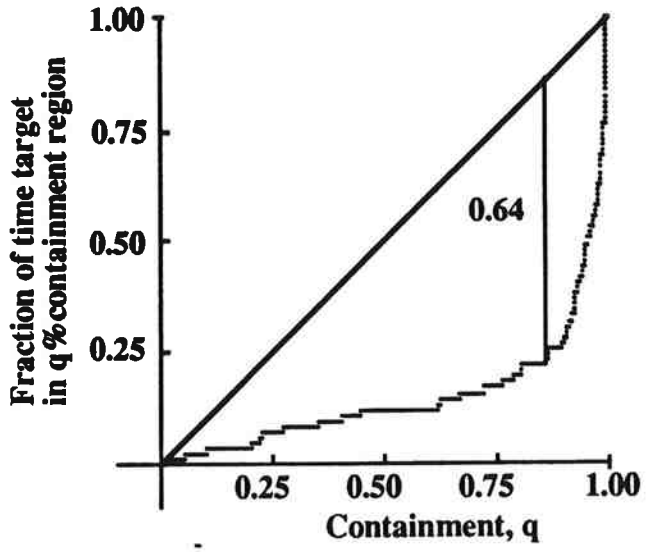
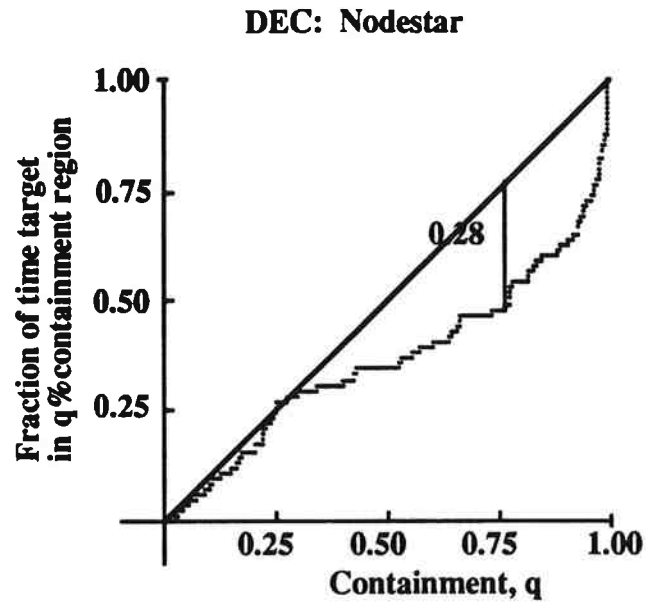


Figure A.14


DEC: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.28
0.25
0.50
0.75
1.00
Containment, q

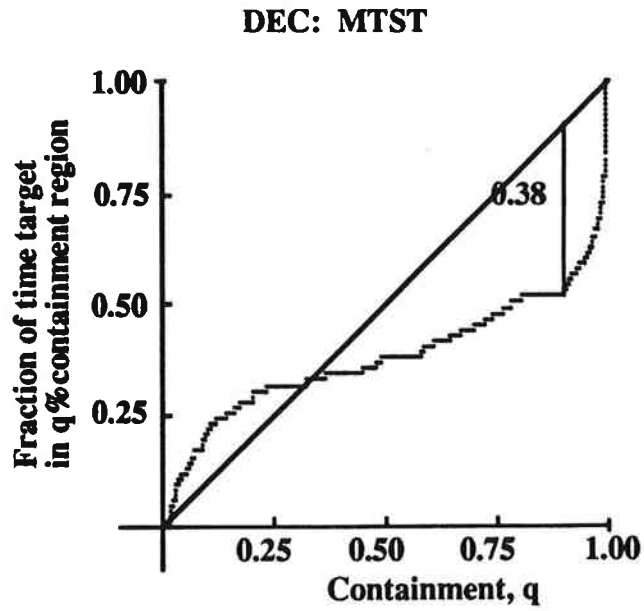
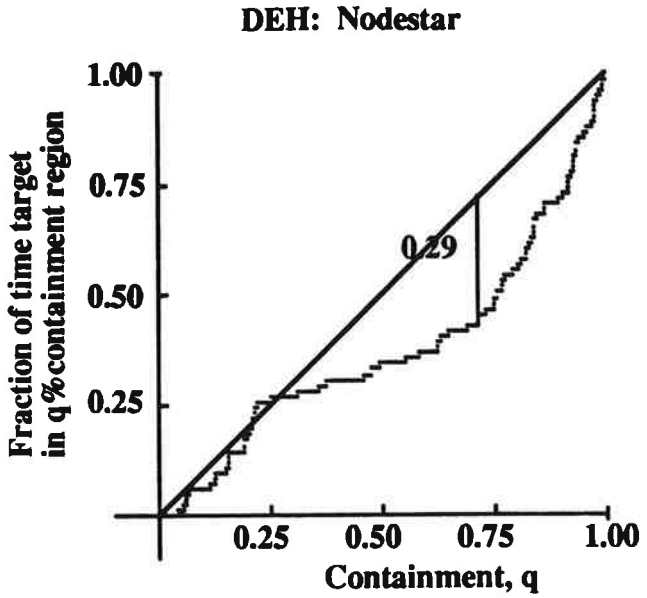

DEC: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.38
0.25
0.50
0.75
1.00
Containment, q

Figure A.15


DEH: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.29
0.25
0.50
0.75
1.00
Containment, q

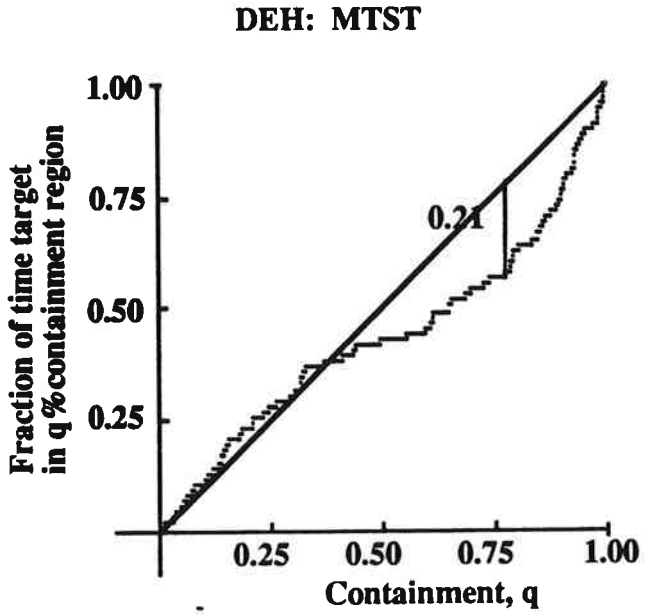
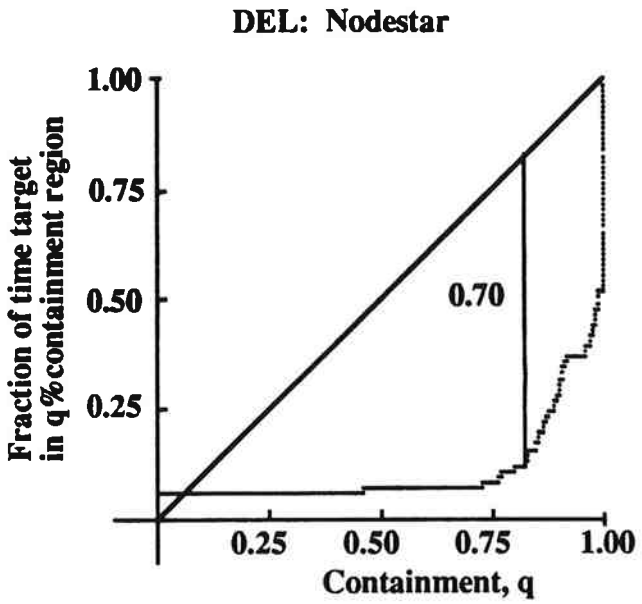

DEH: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.21
0.25
0.50
0.75
1.00
Containment, q

Figure A.16


DEL: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.70
0.25
0.50
0.75
1.00
Containment, q

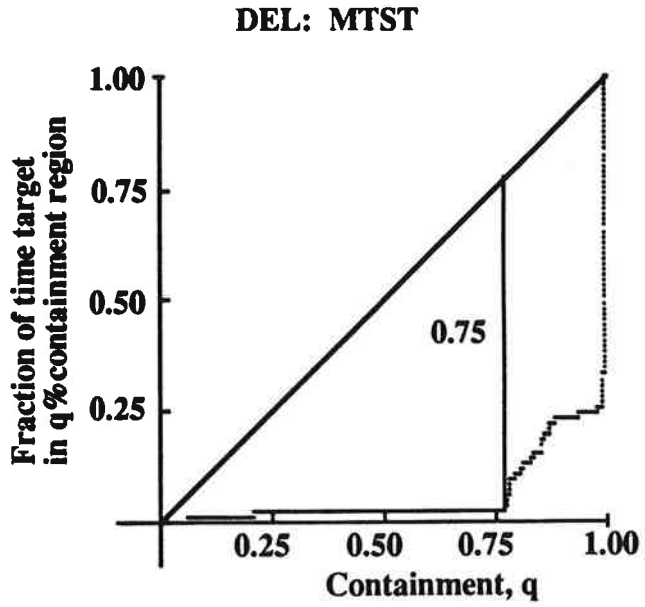
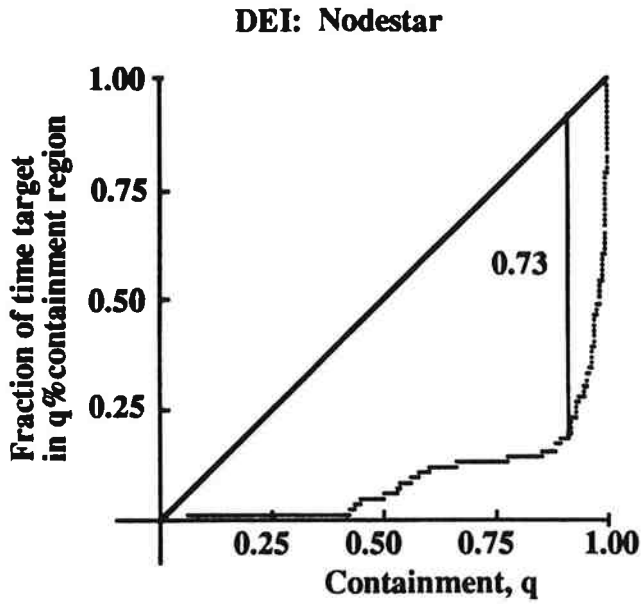

DEL: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.75
0.25
0.50
0.75
1.00
Containment, q

Figure A.17


DEI: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.73
0.25
0.50
0.75
1.00
Containment, q

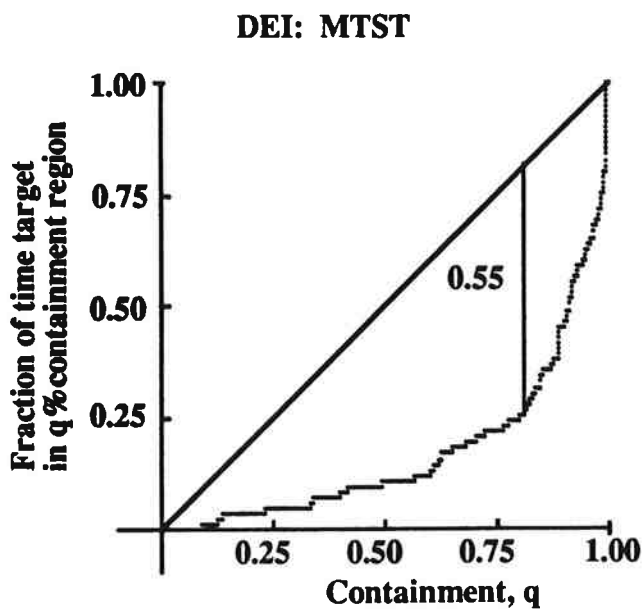
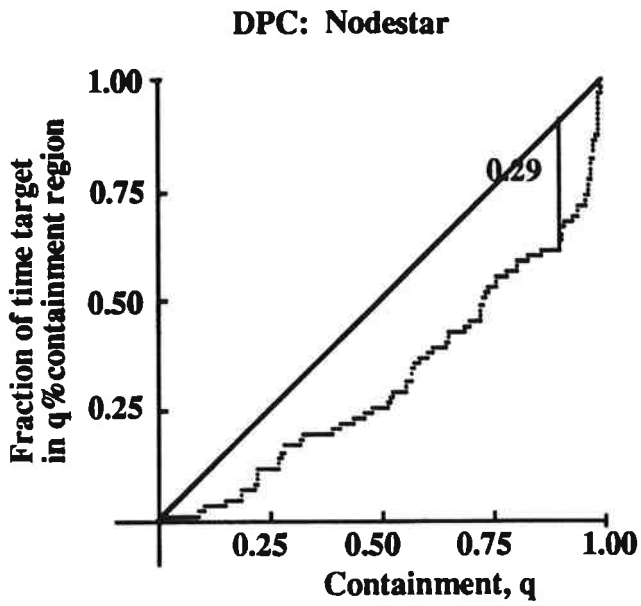

DEI: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.55
0.25
0.50
0.75
1.00
Containment, q

Figure A.18


DPC: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.29
0.25
0.50
0.75
1.00
Containment, q

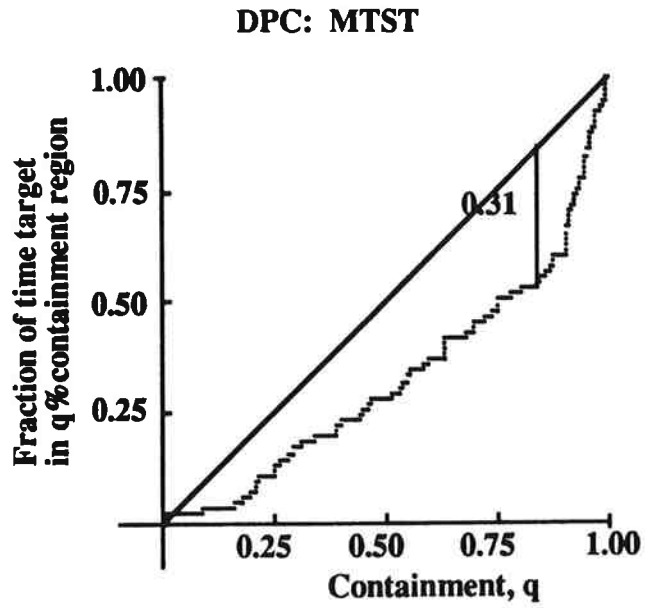
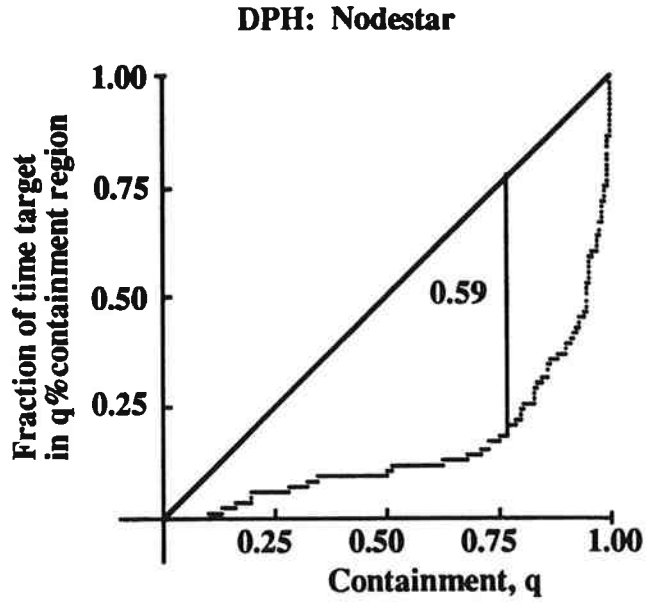

DPC: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.31
0.25
0.50
0.75
1.00
Containment, q

Figure A.19


DPH: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.59
0.25
0.50
0.75
1.00
Containment, q

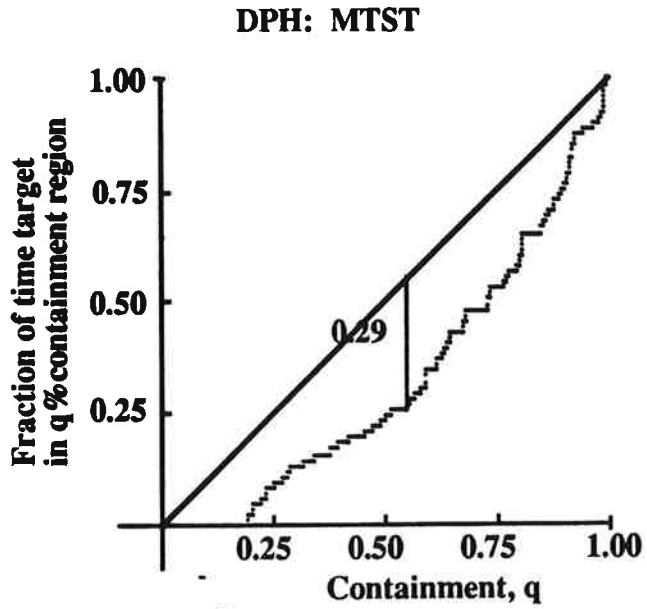
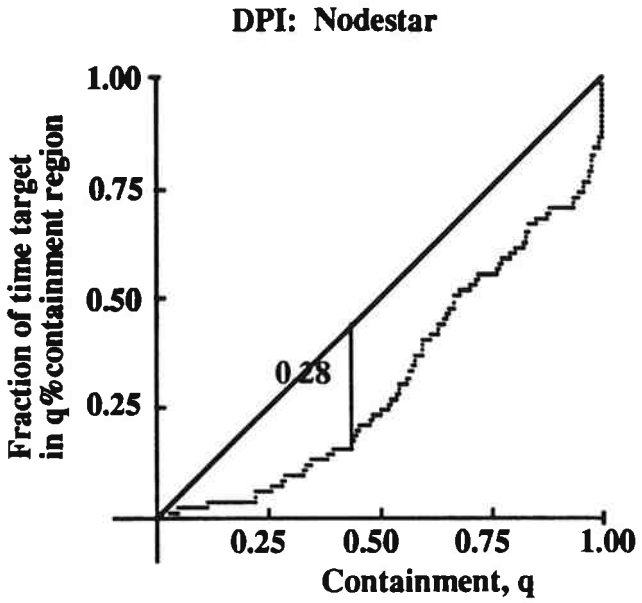

DPH: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.29
0.25
0.50
0.75
1.00
Containment, q

Figure A.20


DPI: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.28
0.25
0.50
0.75
1.00
Containment, q

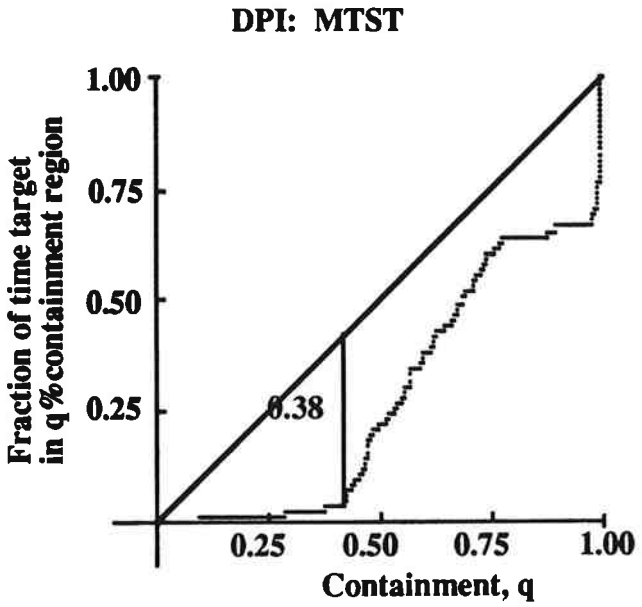
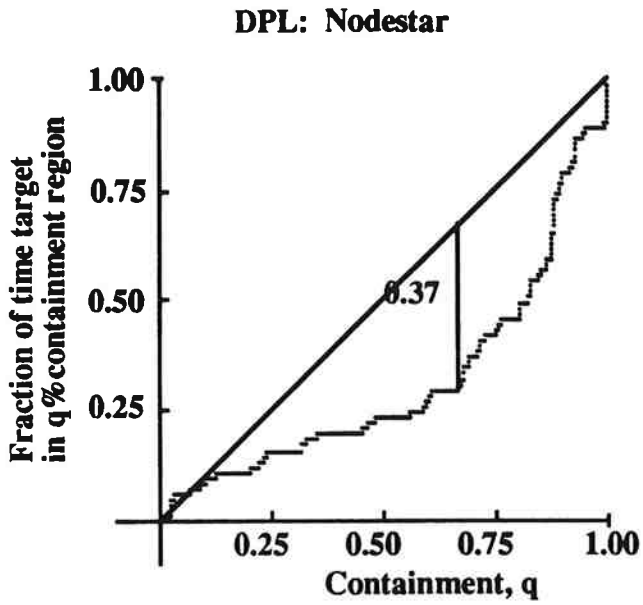

DPI: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.38
0.25
0.50
0.75
1.00
Containment, q

Figure A.21


DPL: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.37
0.25
0.50
0.75
1.00
Containment, q

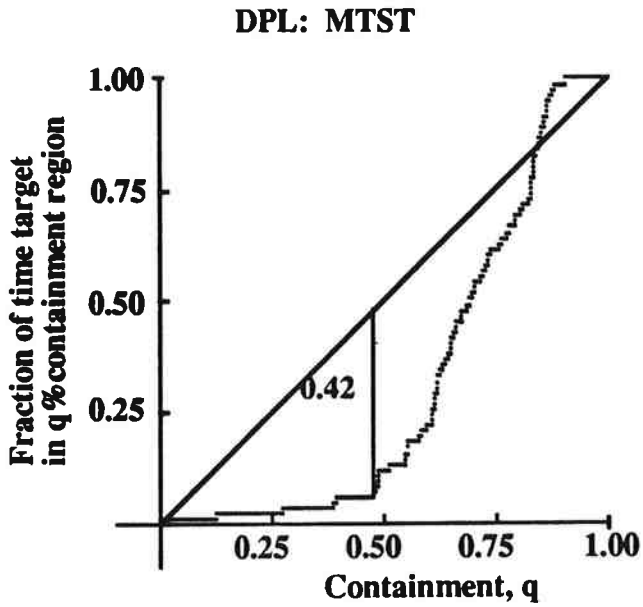
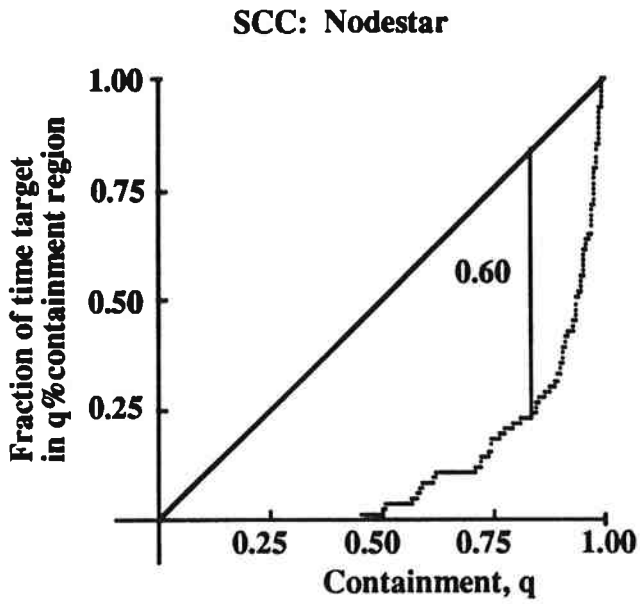

DPL: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.42
0.25
0.50
0.75
1.00
Containment, q

Figure A.22


SCC: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.60
0.25
0.50
0.75
1.00
Containment, q

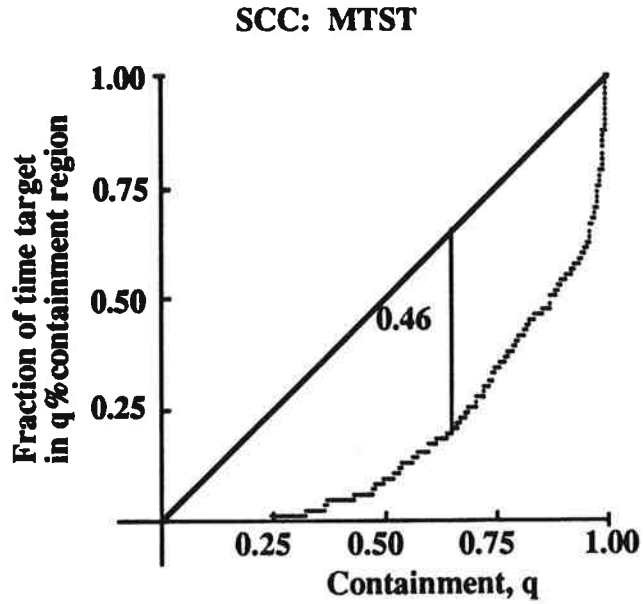
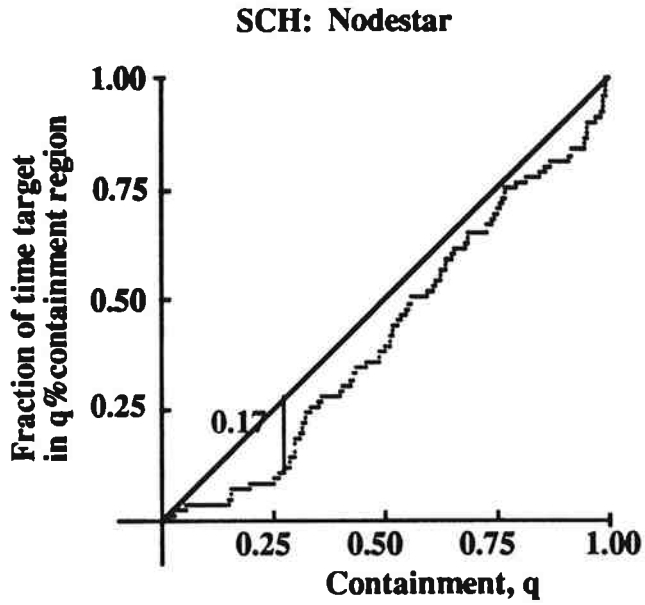

SCC: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.46
0.25
0.50
0.75
1.00
Containment, q

Figure A.23


SCH: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.17
0.25
0.50
0.75
1.00
Containment, q

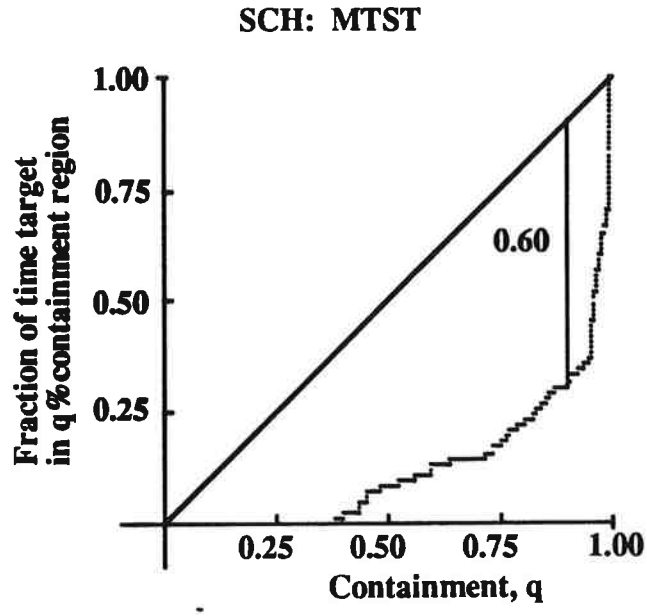
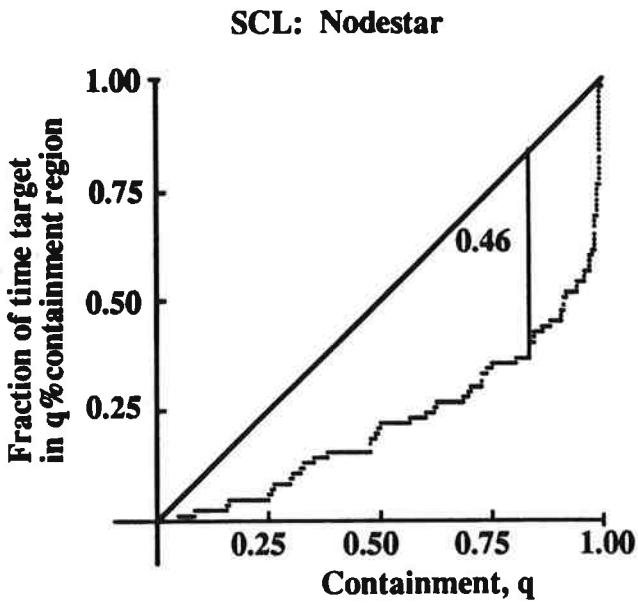

SCH: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.60
0.25
0.50
0.75
1.00
Containment, q

Figure A.24


SCL: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.46
0.25
0.50
0.75
1.00
Containment, q

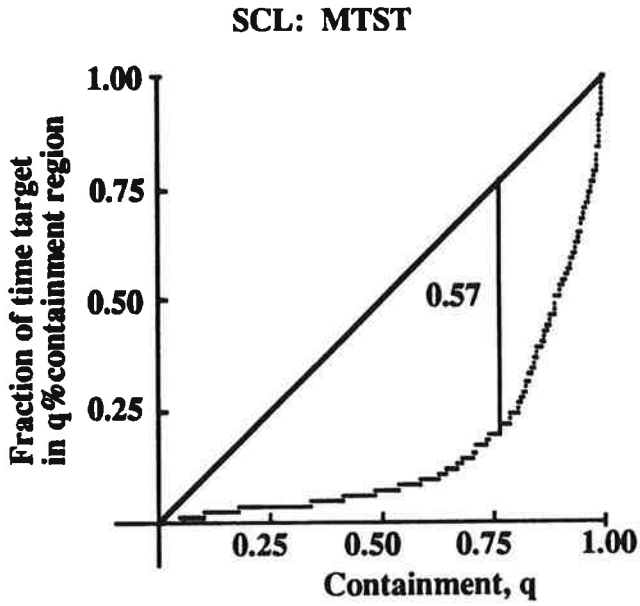
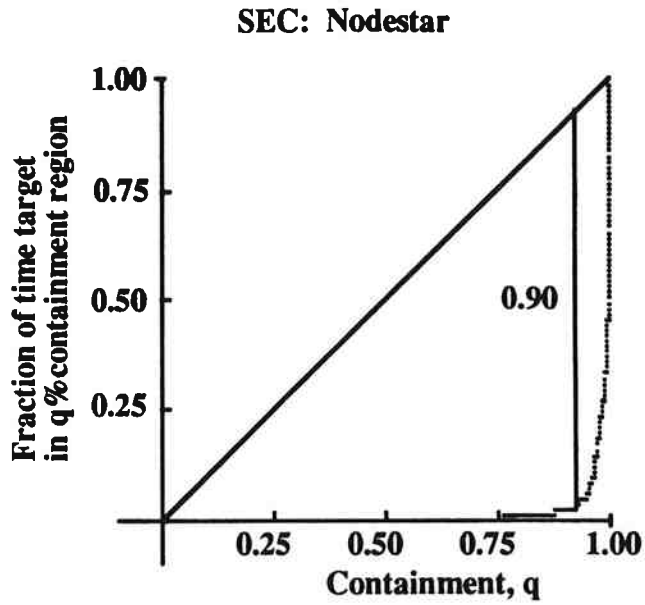

SCL: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.57
0.25
0.50
0.75
1.00
Containment, q

Figure A.25


SEC: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.90
0.25
0.50
0.75
1.00
Containment, q

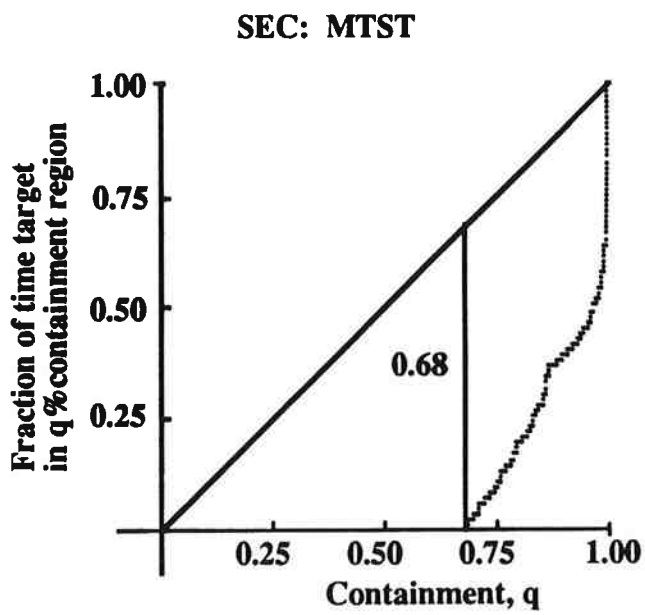
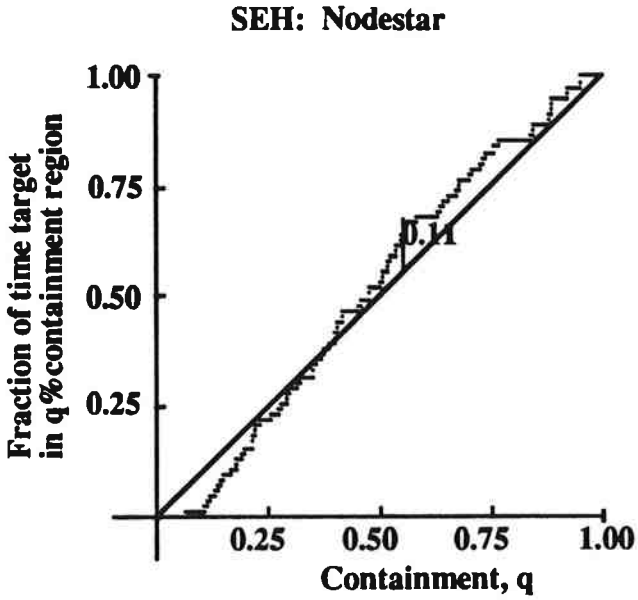

SEC: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.68
0.25
0.50
0.75
1.00
Containment, q

Figure A.26


SEH: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.11
0.25
0.50
0.75
1.00
Containment, q

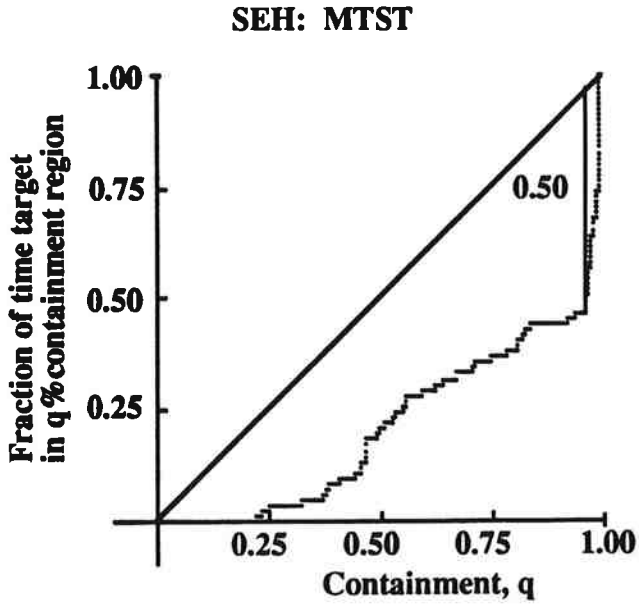
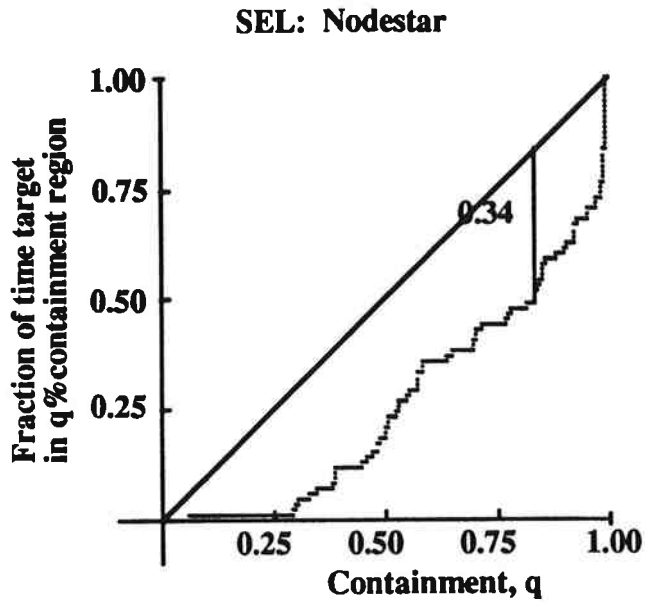

SEH: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q%containment region
0.50
0.25
0.50
0.75
1.00
Containment, q

Figure A.27


SEL: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.34
0.25
0.50
0.75
1.00
Containment, q

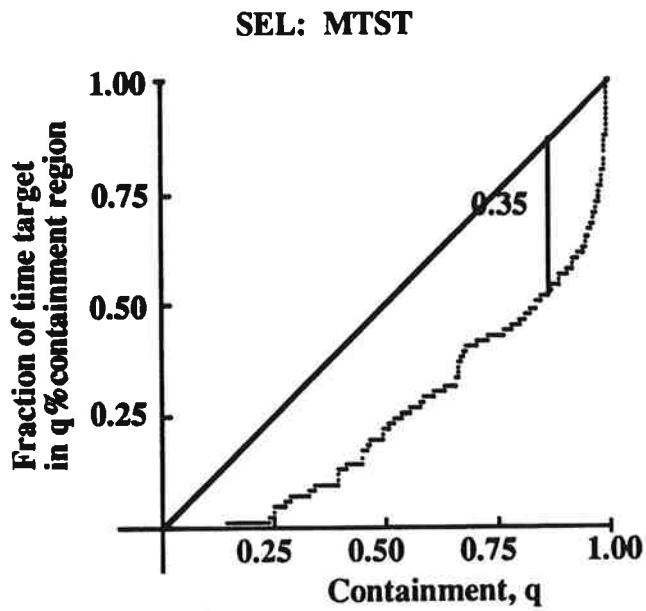
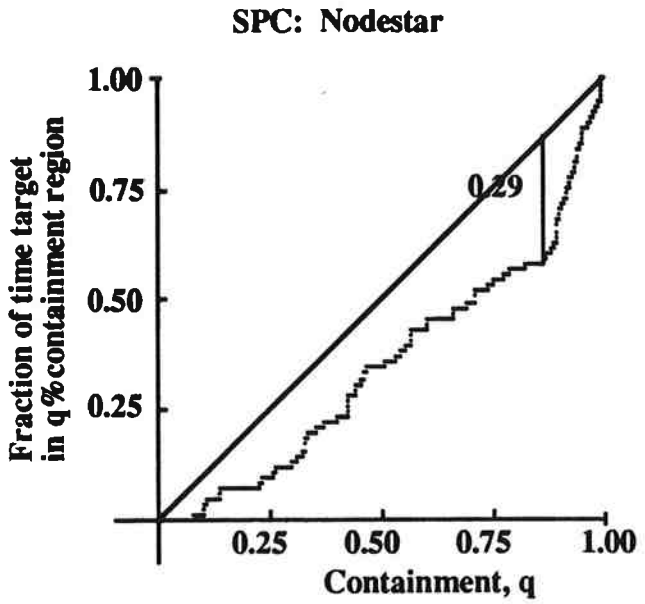

SEL: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.35
0.25
0.50
0.75
1.00
Containment, q

Figure A.28


SPC: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.29
0.25
0.50
0.75
1.00
Containment, q

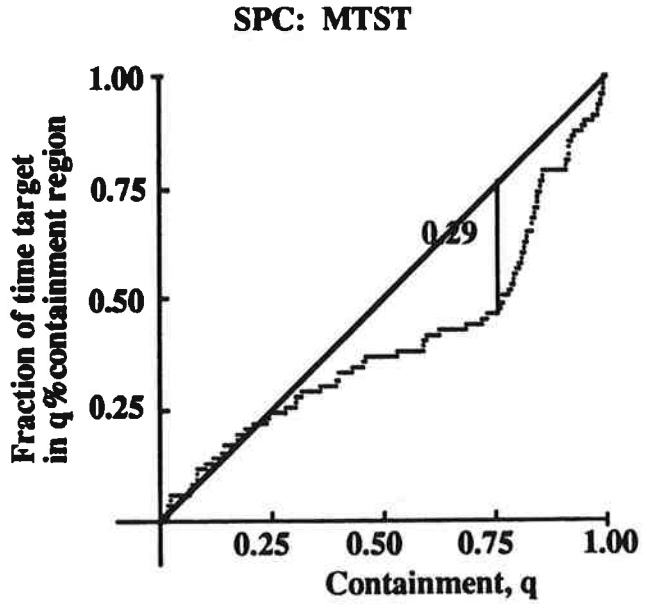
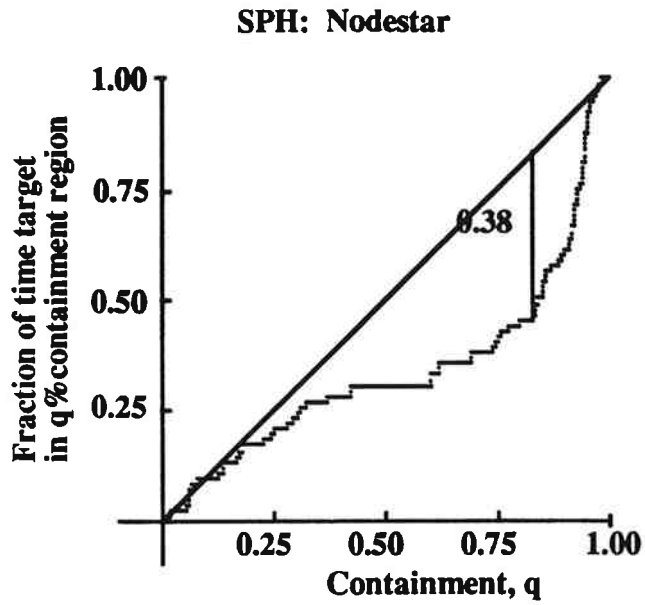

SPC: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.29
0.25
0.50
0.75
1.00
Containment, q

Figure A.29


SPH: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.38
0.25
0.50
0.75
1.00
Containment, q

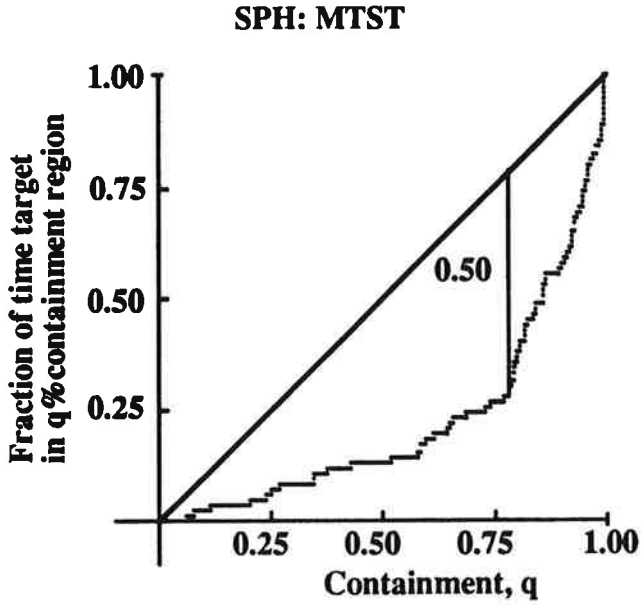
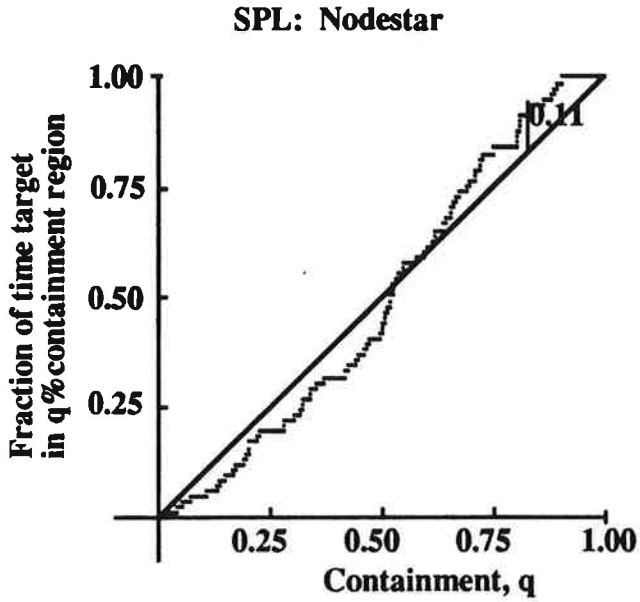
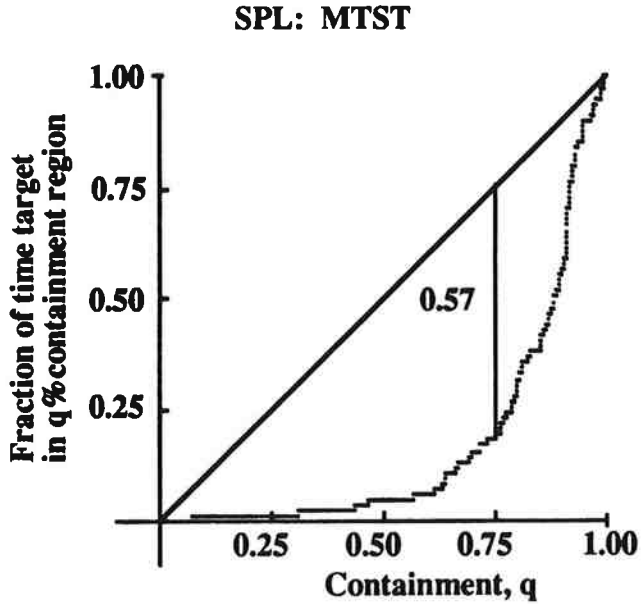

SPH: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.50
0.25
0.50
0.75
1.00
Containment, q

Figure A.30


SPL: Nodestar
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.11
0.25
0.50
0.75
1.00
Containment, q


SPL: MTST
1.00
0.75
0.50
0.25
Fraction of time target in q% containment region
0.57
0.25
0.50
0.75
1.00
Containment, q