Information-based Data Prioritization in Distributed Tracking Systems

Nick Coult[a], J. Nate Knight, Woody Leed, Scott Danford, Randy Paffenroth, and Aubrey Poore

Numerica Corporation, 4850 Hahns Peak Drive Suite 200, Loveland, CO 80538, USA

ABSTRACT

Effective multi-sensor, multi-target, distributed composite tracking requires the management of limited network bandwidth. In this paper we derive from first principles a value of information for measurements that can be used to sort the measurements in order from most to least valuable. We show the information metric must account for the models and filters used by the composite tracking system. We describe how this value of information can be used to optimize bandwidth utilization and illustrate its effectiveness using simulations that involve lossy and latent network models.

Keywords: Kullback-Leibler Divergence, data prioritization, distributed tracking

1. INTRODUCTION

Distributed tracking systems which share track data such as associated measurement reports (AMRs) between multiple nodes on a network typically have a limited resources, including limits on the amount of bandwidth available, as well as limited resources that can be expended by a sensor in illuminating or detecting targets. For example, the desired send rate of data from a particular node on such a network may in fact exceed the maximum send rate supported by the network.

In such a cases, the sending node must *prune* the data. Pruning can be used to limit the data that is transmitted by the node into the network, as well as managing the resources of sensors so that a given sensor will not expend resources on objects for which its local data would be pruned. Both problems are of interest, though we present our results in the context of pruning for network optimization.

Rather than pruning randomly or arbitrarily, the pruning algorithm should prioritize the track data based on the value it provides to network track picture. Schmaedeke and Kastella[1] demonstrated the use of discrimination gain for a related problem in sensor resource management. Others have shown the use of mutual information[2] to optimize sensor coverage with multiple sensors. The discrimination gain is also known as the *Kullback-Leibler divergence* (KLD).[3]

The KLD provides a measure of the difference in information between two probability distributions. Schmaedeke and Kastella showed how to use the KLD for measuring the relative value that multiple sensors provide to a given track. As they demonstrated, in the case of Gaussian distributions, the KLD can be simplified into a sum of four terms. In this paper, we show a useful kinematic interpretation of these terms. Furthermore, we show that for an Interacting Multiple Model Filter, the KLD can be simplified considerably beyond what has been shown in previous work by others. As our numerical experiments show, the KLD *implicitly* and *automatically* takes into account multiple factors which contribute to the value of an AMR, including sensor location and quality, target maneuvers, and time since last update.

This paper is structed as follows. Sections 2.1 and 2.2 set the context of priotization and pruning of AMR data. Section 2.3 shows our intepretation of the KLD for the Kalman filter and the IMM filter. Finally, Section 3 demonstrates the use of the KLD for pruning AMR data in numerical experiments.

To whom correspondence should be addressed: nick.coult@numerica.us, telephone: +1 970 461 2000 x 252

2. PRUNING AND PRIORITIZATION

2.1 Prioritization via scoring

Consider a stream of AMRs $M = \{m_1, m_2 \ldots\}$ emanating from a single tracking node A. (A given AMR m_i consists of an *association* to a track, and kinematic measurement data with timestamp). In a perfect network, all AMRs would be sent and received instantaneously, and recipient nodes $B, C, D...$ could have an exact copy of the track picture that exists at A.

Real networks have limited bandwidth. Therefore, in order to avoid saturating the network, A must choose a subset of M to transmit based on the available bandwidth. In particular, we consider the following constraint: given a sequence of *transmit opportunities* at times $t_1, t_2 \ldots$, the network has bandwidth available for the transmission of a single AMR.

Thus, at time t_k, A must decide which of the AMRs that have been generated up to time t_k, *but which have not yet been transmitted*, should be transmitted at the current opportunity. Inherently, A has a queue of untransmitted AMRs stored up. At each transmit opportunity, it chooses one AMR from this queue to send.

Furthermore, the AMR queue can be represented in a general fashion using a *priority queue*. Each entry in a priority queue has a numerical *score*, and the highest-scoring entry of the queue is chosen first. The use of a priority queue to store the untransmitted AMRs boils the problem space down to the choice of a function to compute the score of a given AMR. Different scoring functions will affect the received track picture in different ways at $B, C, D....$ The particular function to use depends on the desired metrics to optimize at the recipient nodes.

Let us formalize the notation a bit. Consider a set of AMRs $\tilde{M} \subset M$ that has already been transmitted, and a set of AMRs $\bar{M} = M \setminus \tilde{M}$ that has not been transmitted. The elements of $\bar{M}$ can be prioritized from highest to lowest score by defined a scoring function $f : \bar{M} \to \mathbb{R}$ such that for any given $\bar{m} \in \bar{M}$, $f(\bar{m})$ is a function of the track attributes that can be derived from the set of AMRs $\tilde{M} \cup \{\bar{m}\}$ relative to the track attributes that can be derived from the set of AMRs $\tilde{M}$. In other words, we score $\bar{m}$ by considering the track picture both *with* and *without* the AMR $\bar{m}$. Thus, given n AMRs in $\bar{M}$, the priority queue $\Pi(\bar{M})$ that can be derived from $\bar{M}$ and f is of the form

$$\Pi(\bar{M}) = \{(\bar{m}_i, f(\bar{m}_i)\}_{i=1}^{n}.$$

At this point, there is a subtle wrinkle in the formulation that has important computational implications: $\Pi(\bar{M})$ is *not* a subset of $\Pi(\bar{M} \cup \{\bar{m}\})$, because the score of some or all elements of $\bar{M}$ could change if $\bar{m}$ is actually transmitted. The computational implication is that without further constraints or assumptions, one has to re-score the entire priority queue whenever a single AMR is transmitted. For systems producing AMRs at a high rate, but with few transmit opportunities, this could get computationally expensive.

In our view, there is minimal impact to further constrain the priority queue so that re-scoring is not required. In other words, once $f(\bar{m})$ is evaluated for a given AMR $\bar{m}$, it is not re-evaluated.

2.2 Pruning

If the transmit opportunities t_k persistently occur at a lower rate than AMRs become available, then the network link becomes oversubscribed and the size of $\bar{M}$ could grow without bound. This is where the benefit of the prioritization scheme outlined previously becomes most apparent. (Note that if the network link is only transiently oversubscribed, then the prioritization acts as a *traffic shaper* by smoothing out the transmit rate, but all AMRs are still eventually sent. The effects on the track picture in such a case are not the subject of this paper).

In order to prevent the unbounded growth of $\bar{M}$, once an AMR $\bar{m} \in \bar{M}$ reaches a predetermined age, it should be pruned from $\bar{M}$ in order to prevent transmitting it. The reasoning of this additional step is that we assume the scoring function f would, if re-evaluated on older AMRs, return a very low score for them. Since the scores of existing AMRs in $\bar{M}$ are not re-evaluated, there is no way to account for this other than by including a pruning function that is explicitly time-based.

2.3 The Kullback-Leibler Divergence

First introducted by Kullback and Leibler in their 1951 paper,[3] the Kullback-Leibler divergence provides a measure of the difference between two probability distributions. For density functions p and q defined on the space Ω, the general form of the KLD is given by the formula

$$D_{KL}(p,q) = \int\limits_{\Omega} p(x) \log \frac{p(x)}{q(x)}\, dx \tag{1}$$

Strictly speaking, D_{KL} is not a distance metric because $D_{KL}(p,q) \neq D_{KL}(q,p)$. In the context of information theory, the KLD measures the additional number of bits required to code a message drawn from the distribution P when using a code that is optimal for the distribution Q relative to the number of bits required when using a code that is optimal for P.[4] So, if P has more or different information relative to Q, then we expect $D_{KL}(p,q)$ to measure both how much additional information is present in P, and how different the information of P is than Q.

The KLD can be used directly as a scoring function as follows:

1. For a given AMR $\bar{m} \in \bar{M}$ associated to a particular track, compute two different track states:
   (a) First, compute the track state of the track using $\bar{m}$ and any other AMRs from $\bar{M}$ associated to the track; let the associated track state distribution be P.
   (b) Then, propagate the track state to the time of $\bar{m}$ but do *not* update the track with $\bar{m}$. Let the associated track state distribution be Q.
2. $f(\bar{m}) = D_{KL}(P,Q)$.

KLD and Normal Distributions. When p and q are normal densities with means μ_p, μ_q, covariances Σ_p, Σ_q, and dimension k, (1) can be simplified further to

$$D_{KL}(p,q) = \frac{1}{2}\left((\mu_p - \mu_q)^T \Sigma_Q^{-1} (\mu_p - \mu_q) + \mathrm{tr}(\Sigma_\mathrm{Q}^{-1}\Sigma_\mathrm{P}) - \log\left(\frac{|\Sigma_\mathrm{P}|}{|\Sigma_\mathrm{Q}|}\right) - \mathrm{k}\right). \tag{2}$$

Since many tracking systems make use of Gaussian distributions to represent track states, the formula (2) may be used for computing the score. In this context, the three terms in that formula can be interpreted in both kinematic and information-theoretic terms:

- $(\mu_p - \mu_q)^T \Sigma_Q^{-1} (\mu_p - \mu_q)$ measures the difference (relative to the predicted covariance before update) between the predicted and updated track state mean. If $\bar{m}$ represents an AMR on a maneuvering track, then this term would be larger than if the track is not maneuvering (in which case the prediction is more likely to be close to the updated value). This term is a therefore a measure of the *information difference* between P and Q. Even if P and Q have exactly the same covariance, P may be providing different information because the mean is different.
- $\log\left(\frac{|\Sigma_P|}{|\Sigma_Q|}\right)$ measures the relative change in the volume of the covariance ellipsoid. If P has a much smaller covariance than Q, this term is smaller than if the covariance of Q is similar in magnitude to the covariance of P. Thus, an AMR $\bar{m}$ which greatly reduces the covariance causes the score of $\bar{m}$ to be larger.(We also note that for a Kalman filter,$|\Sigma_P| < |\Sigma_Q|$ regardless of the measurement.) So, we can see that this term is a measure of the *additional information* present in P compared to Q. Even if the mean is unchanged, P may have a smaller covariance and therefore be providing more information than Q.
- The term $\mathrm{tr}(\Sigma_\mathrm{Q}^{-1}\Sigma_\mathrm{P})$ is more subtle. We consider two cases:

1. $|\Sigma_Q| \approx |\Sigma_P|$. In this case, $|\Sigma_Q^{-1}\Sigma_P| = \frac{|\Sigma_P|}{|\Sigma_Q|} \approx 1$. Since the determinant is the product of the eigenvalues, for every eigenvalue of $\Sigma_Q^{-1}\Sigma_P$ which is close to zero, there must be an eigenvalue which is very large. In order for $\Sigma_Q^{-1}\Sigma_P$ to have an eigenvalue close to zero, there must be a subspace on which $(\Sigma_P x, x)$ is close to zero and $(\Sigma_Q x, x)$ is not close to zero (under the constraint $\|x\| = 1$), or vice versa.

   We can interpret this to mean that $\mathrm{tr}(\Sigma_\mathrm{Q}^{-1}\Sigma_\mathrm{P})$ is large only if the covariance ellipsoids of Σ_P and Σ_Q *do not have the same shape*. If the shape is similar, then the contribution of this term will be moderate. In other words, this term is measuring information difference even if the amount of information is roughly the same.

2. $|\Sigma_P| \ll |\Sigma_Q|$. In this case, $|\Sigma_Q^{-1}\Sigma_P| \approx 0$. However, this does not preclude some eigenvalues of $\Sigma_Q^{-1}\Sigma_P$ from being large (and therefore $\mathrm{tr}(\Sigma_\mathrm{Q}^{-1}\Sigma_\mathrm{P})$ could still be large). By a recursive argument, we can show that either there is a subspace on which condition 1 above applies, or there is no such subspace. If there is no such subspace, then under the assumption $|\Sigma_P| \ll |\Sigma_Q|$ on all subspaces, the eigenvalues of $\Sigma_Q^{-1}\Sigma_P$ must be uniformly close to zero.

We have shown that when $\mathrm{tr}(\Sigma_\mathrm{Q}^{-1}\Sigma_\mathrm{P})$ contributes to the KLD, it is measuring the difference in information due to the change in shape of the covariance ellipsoid.

KLD and Interacting Multiple Model Filters.

The Interacting Multiple Model[5] (IMM) generalizes the concept of a Kalman filter. Given r motion models each defined on a state space Ω, the IMM propagates and updates r filters, each with a modal probability μ_i. So, we can view the state space of the IMM as a hybrid discrete-continous space $\Gamma = \Omega^r$

Now, consider (1) evaluated on Γ, where

$$p(x) = (\mu_1^p p_1(x_1), \ldots, \mu_r^p p_r(x_r)))$$

and $q(x)$ is defined similarly.

So, we have:

$$\begin{aligned} D_{KL}(p,q) &= \int\limits_{\Gamma} p(x) \log \frac{p(x)}{q(x)}\, dx & (3) \\ &= \sum_{i=1}^{r} \int\limits_{\Omega} \cdots \int\limits_{\Omega} \mu_i^p p_i(x_i) \log \frac{\mu_i^p p_i(x_i)}{\mu_i^q q_i(x_i)}\, dx_1 \cdots dx_r & (4) \\ &= \sum_{i=1}^{r} \int\limits_{\Omega} \cdots \int\limits_{\Omega} \mu_i^p p_i(x_i) \left(\log \frac{\mu_i^p}{\mu_i^q} + \log \frac{p_i(x_i)}{q_i(x_i)} \right) dx_1 \cdots dx_r & (5) \\ &= \sum_{i=1}^{r} \mu_i^p \left(\log \frac{\mu_i^p}{\mu_i^q} + D_{KL}(p_i, q_i) \right) & (6) \end{aligned}$$

In the final formula (6), we that the updated IMM filter represented by p is scored against q using a sum of terms of the form $\log \frac{\mu_i^p}{\mu_i^q} + D_{KL}(p_i, q_i)$ where the weights in the sum are the updated modal probabilities. The term $\log \frac{\mu_i^p}{\mu_i^q}$ measures how much the modal probability i has changed; in particular, if μ_i^p is much larger than μ_i^q then this term will contribute. The term $D_{KL}(p_i, q_i)$ is the standard KLD applied to the individual filter states p_i and q_i.

So, in summary the KLD effectively prioritizes increases and changes in information, which for IMM filters amounts to the following:

- Difference in predicted versus updated mean.

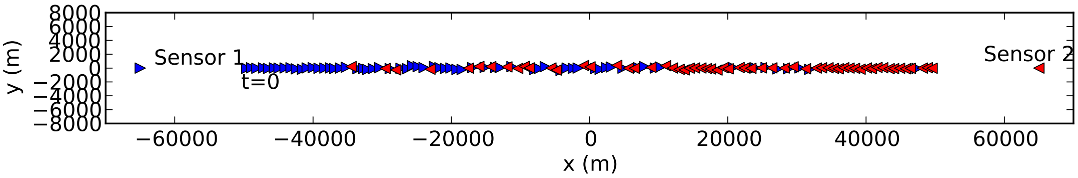

Figure 1. Straight line target flying between two range-bearing sensors. Each sensor generates an AMR every two seconds. The network bandwidth allows for sending only one; the other is immediately pruned. The symbols in the figure represent which sensor's AMR was transmitted at each track update based on the KLD scores. Since each sensor is a range-bearing sensor, as the target moves away from sensor 1 and approaches sensor 2, the KLD scores shift from valuing sensor 1 AMRs to valuing sensor 2 AMRs.

- Decrease in the determinant of the covariance.
- Difference in covariance ellipsoid shape.
- Difference in IMM modal probabilities

3. NUMERICAL EXAMPLES

3.1 Multiple sensors

First, we consider a simple scenario with one target and two sensors, as shown in Figure 1. The sensors are both range-bearing sensors, so that the effective spatial resolution of each sensor depends on the range to the target.

An AMR is generated by each sensor once every 2 seconds, and we have only one transmit opportunity every 2 seconds. Furthermore, for the sake of simplicity, any AMRs which are not transmitted are pruned immediately. Thus, exactly half of the AMRs are transmitted.

We filter using an NCV extended Kalman filter, and score using (2). Our analysis in Section 2.3 suggests that as the target moves away from Sensor 1 and towards Sensor 2, the Sensor 1 AMRs should be pruned more often and the Sensor 2 AMRs less often; the reason is that Sensor 2 will contribute more strongly to the shrinkage in the covariance than Sensor 1. Additionally, we do not expect the change in mean or the change in covariance ellipsoid shape to contribute significantly. Figures 1 and 3 show that this is precisely what happens. As shown previously,[1] we see that the KLD score can be used for *sensor resource management*, since it can automatically determine the value that a sensor provides to a given track.

3.2 Single sensor

Now, we consider a scenario with only one sensor, but six targets. The targets begin the scenario in straight-line motion, after which one of the targets peels off in a high G turn. The target motion is shown in Figure 4. Every two seconds, the sensor generates an AMR on all six targets simultaneously. As shown in Figure 5, the KLD score is much higher for the maneuvering target during the turn.

Figure 6 shows the same scenario, but with only one out of six AMRs transmitted. Each group of six AMRs is scored using the KLD, and only the highest-scoring AMR is transmitted. Before the turn, the update rate of the targets is fairly uniform. As a given target goes longer without getting an update, the covariance

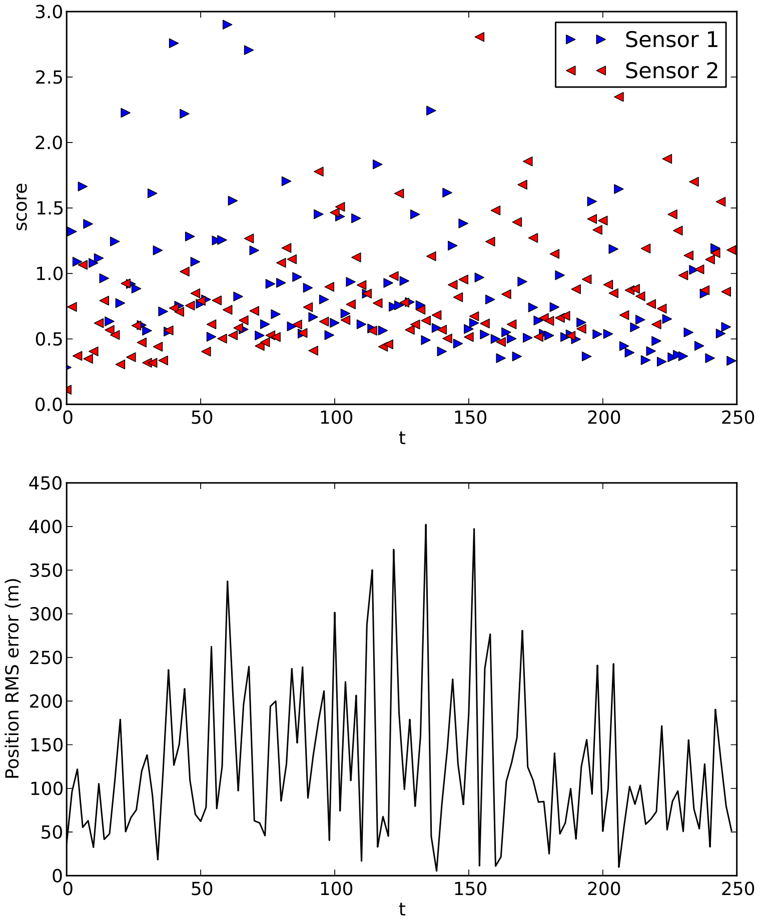

Figure 2. KLD scores and RMS position error for the scenario shown in Figure 1.

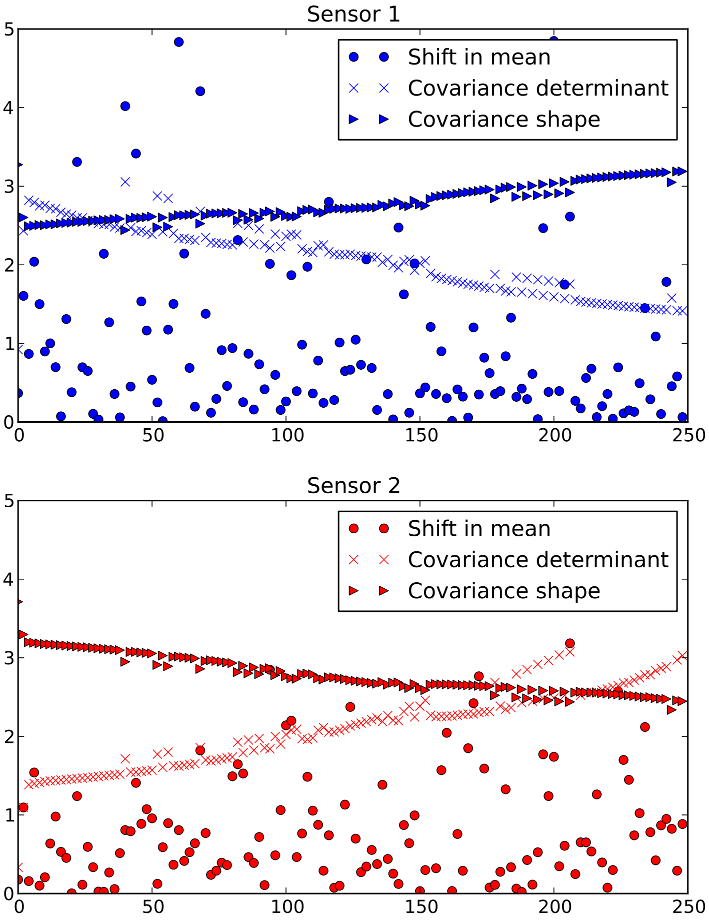

Figure 3. Individual KLD scoring terms for the scenario shown in Figure 1. The dominant term is $\log\left(\frac{|\Sigma_P|}{|\Sigma_Q|}\right)$ due to the sensor and target geometry.

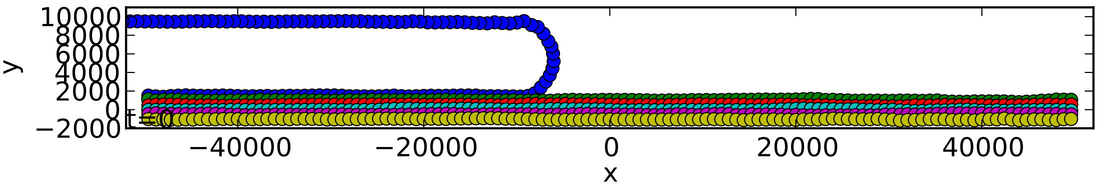

Figure 4. Target motion. The colored circles represent the AMR's that are available to transmit.

shrinkage term in the KLD score grows until eventually that track scores the highest and gets updated. Once the maneuver begins, the maneuvering target achieves a higher effective update rate than the other targets during the maneuver because the IMM weight term and the prediction error term dominate. So, we see that the KLD score provides a means of automatically identifying the important information on a track for both non-maneuvering and maneuvering targets.

For comparison, consider Figures 8 and 9, which also show the results of sending only one out of every six AMRs; however, instead of using the KLD score to prioritize, each track gets updated at an equal rate using exactly one sixth of the AMRs available. The position and velocity error is significantly degraded.

REFERENCES

[1] Schmaedeke, W. and Kastella, K., "Sensor management using discrimination gain and interacting multiple model kalman filters," *IMA Preprint Series* **1555** (April 1998).

[2] Grocholsky, B., Durrant-whyte, H., and Gibbens, P., "Information-theoretic approach to decentralized control of multiple autonomous flight vehicles," in [*In Proceedings of SPIE Vol. 4196, Sensor Fusion and Decentralized Control in Robotic Systems III*], 348–359 (2000).

[3] Kullback, S. and Leibler, R., "On information and sufficiency," *Annals of Mathematical Statistics* **22**(1), 79–86 (1951).

[4] Cover, T. M. and Thomas, J. A., [*Elements of information theory*], Wiley-Interscience, New York, NY, USA (1991).

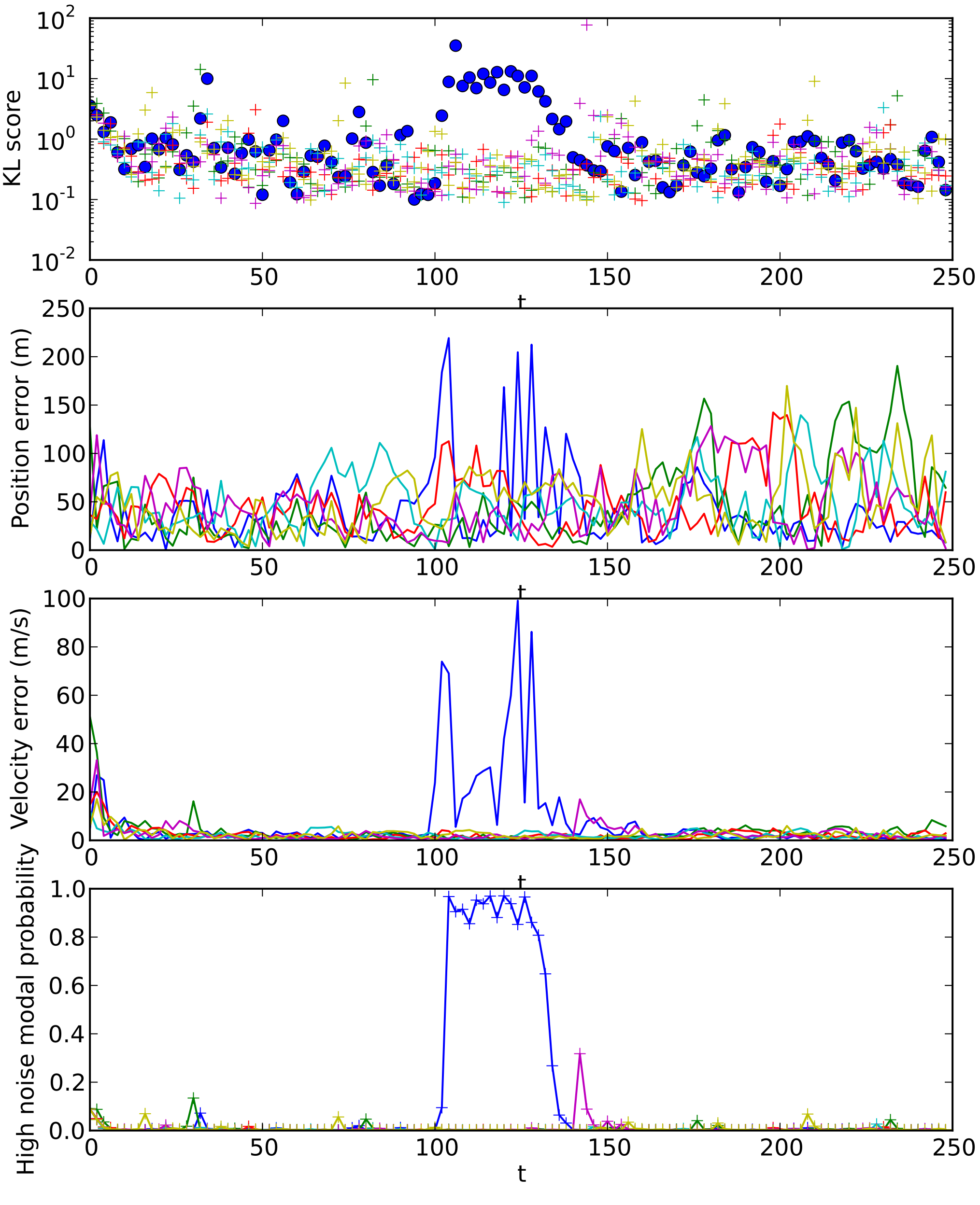

Figure 5. Metrics on the scenario shown in Figure 4. The blue circles in the KLD score represent the KL scores of the maneuvering target. The KLD score is substantially higher during the manuever.

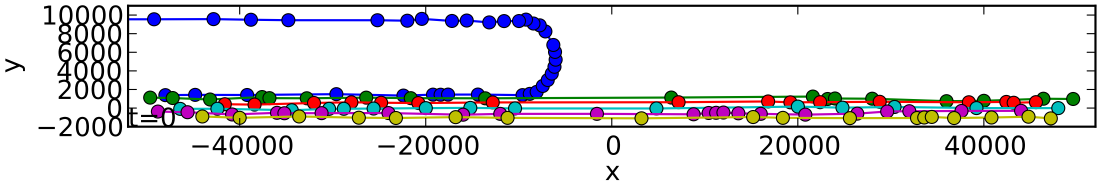

Figure 6. Same scenario as in Figure 4, but only one out of every six AMR is transmitted. The KLD is used to score each group of six AMRs, and the highest scoring from each group is transmitted; the remainder are discarded. The high score on the maneuvering target during the turn results in a higher update rate on that target.

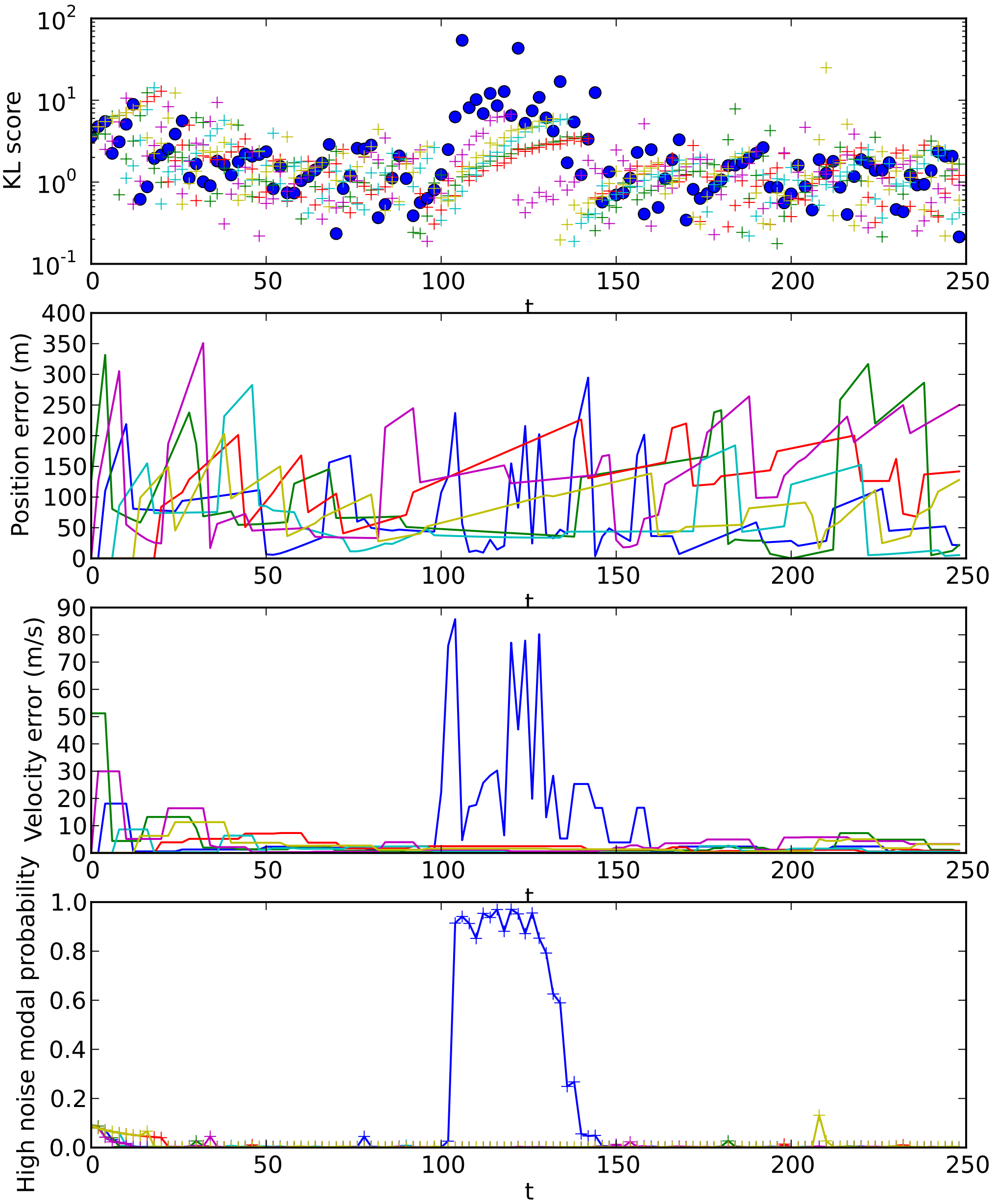

Figure 7. Metrics on the scenario shown in Figure 6. The position accuracy degrades slightly on all targets, but a fairly uniform bound is maintained on all targets including the maneuvering target.

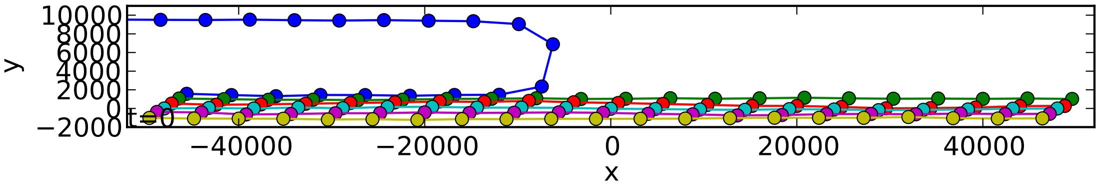

Figure 8. Same scenario as in Figure 4, but only one out of every six AMR is transmitted. Instead of scoring via the KLD, each target gets updated at an equal rate.

[5] Bar-Shalom, Y., Li, X.-R., and Kirubarajan, T., [*Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*], J. Wiley (2001).

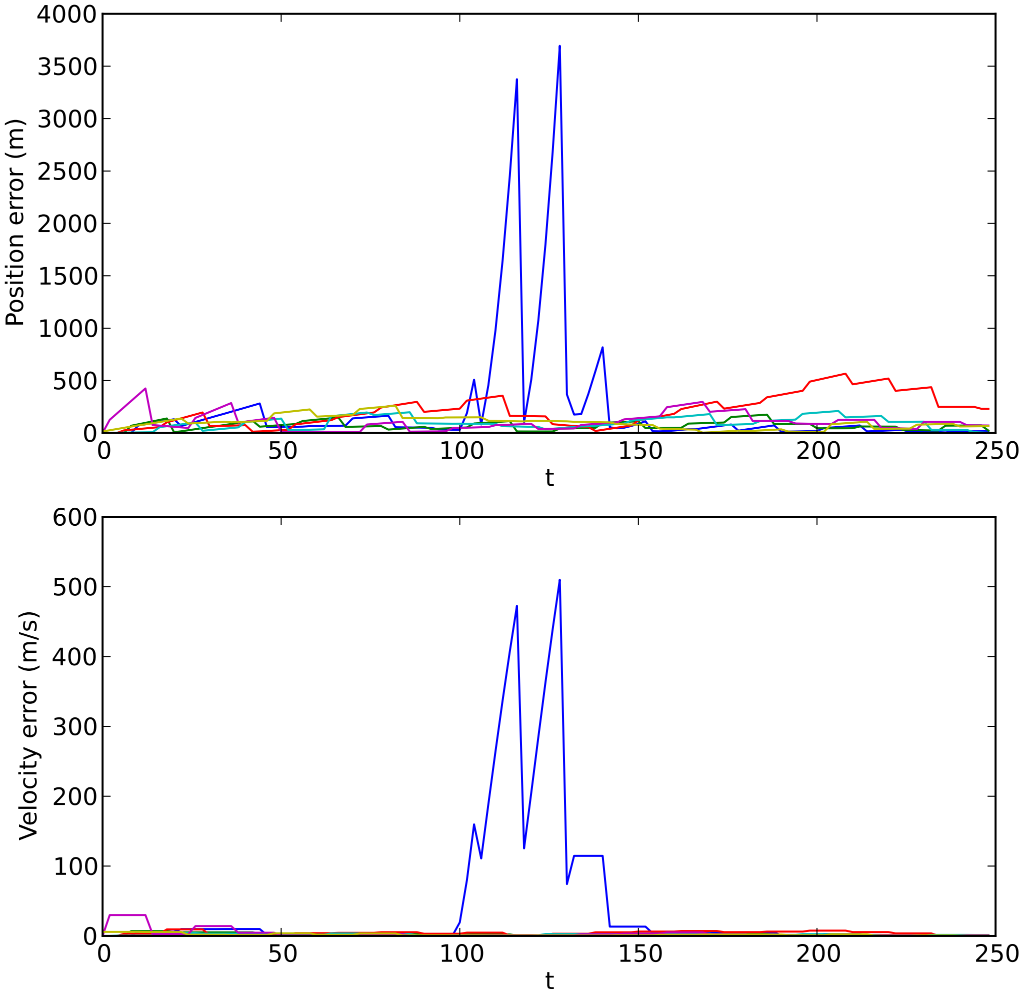

Figure 9. Metrics on the scenario shown in Figure 8. Position and velocity error are degraded significantly during the maneuver.