# On Distributed Processing for Underwater Cooperative Localization

Gao Rui and Mandar Chitre
ARL, Tropical Marine Science Institute, National University of Singapore

***Abstract*—Due to the limited bandwidth of underwater communication links, underwater cooperative localization usually adopts a distributed processing architecture. Members of the team estimate positions using their local sensor data, and fuse the information communicated by other members for cooperation. It is common practice to naïvely assume independency during information fusion between cooperative members. The assumption is not always valid. This results in overconfidence in estimation as a result of the double-counting of the common information. While this problem is recognized by many researchers, there has been no explicit study on the dangers of naïve filtering in presence of inter-dependency. In this paper, we derive an optimal fusion for distributed cooperative localization in a multi-sensor tracking application, and evaluate its gap with respect to the central filtering. For the naïve filtering, we examine one-step and asymptotic performance and demonstrate the existence of safe and dangerous regions of operation.**

***Keywords*—Distributed estimation, cooperative tracking, cooperative localization, marine robots.**

## 1. Introduction

There are two types of cooperative localization problems: the multi-sensor tracking problem and the multi-vehicle localization problem. The multi-sensor tracking problem consists of a team of cooperative nodes tracking a common target. The multi-vehicle localization problem consists of a team of cooperative vehicles estimating their own positions. In both problems, information is shared across the team and improves the estimation as compared to a single sensor localization (without cooperation). If all local sensor data are accessible to a central node, central filtering (CF) yields the optimal estimation. However, underwater wireless (acoustic) communication usually has limited bandwidth, high packet loss and long latency, making it difficult to implement CF in practice. The size of underwater transmission packets is also constrained, and therefore the information shared between members are the processed estimates rather than raw sensor data history. Distributed filtering (DF), where members only process their local sensor data and information communicated by others, is commonly used instead. It is desired that the cooperation to fuse local and remote information outperforms the single filtering (SF) without information sharing.

We focus on the multi-sensor tracking problem. We use the CF and SF as the performance benchmarks. To illustrate the key ideas, we present a simplified problem where two sensor nodes are tracking a common target. While the problem formulation is quite general, we demonstrate most of our results in one dimension and answer the following questions:

- What is the optimal DF?
- Is there any gap between the optimal DF and CF? If so, what it is?
- Is the naïve filtering (NF), where independency between cooperative information is assumed, always acceptable? When does it fail?

These answers give us a clear understanding about the pros and cons of implementing DF under naïve assumptions.

The paper is organized as follows: Section 2 formulates the simplified problem for multi-sensor tracking, and states the implementation and results of SF and CF. Section 3 proposes DF, derives an optimal fusion, and evaluates its performance with respect to CF. Section 4 examines the one-step and asymptotic performance of NF, and derives dangerous regions where NF fails. Section 5 provides a summary of the work, and outlines our future work on multi-vehicle localization.

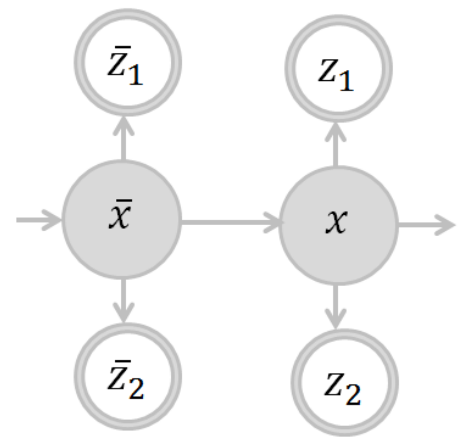



Fig. 1. Multi-sensor tracking: a recursive two-step flow chart.

## 2. Multi-Sensor Tracking

Fig. 1 shows a recursive two-step process where two sensor nodes (node 1 and 2) track a target with true position state $\mathbf{x}$ of size $n\times 1$. Variables with a bar on the top are at the previous step while the ones without a bar are at the current step. The target propagates from previous position $\bar{\mathbf{x}}$ to current position $\mathbf{x}$ with some propagation noise $\mathbf{w}$. At each step, each node makes an observation ($\mathbf{z}_1$ or $\mathbf{z}_2$) about the target position. The observations are independent of each other and independent from the propagation noise of the target. The propagation model and observation models are:

$$\begin{aligned} \mathbf{x} &= \bar{\mathbf{x}} + \mathbf{w}, \\ \mathbf{z}_1 &= \mathbf{x} + \mathbf{v}_1, \\ \mathbf{z}_2 &= \mathbf{x} + \mathbf{v}_2, \end{aligned} \tag{1}$$

where the propagation noise $\mathbf{w}$ and observation noises $\mathbf{v}_1$ and $\mathbf{v}_2$ are independent zero-mean Gaussian processes with covariances $\mathbf{Q}$, $\mathbf{R}_1$ and $\mathbf{R}_2$ respectively. While it is easy to extend this formulation to allow more complex propagation and measurement models, we keep the models intentionally simple to illustrate the basic ideas with minimal mathematical complexity. The problem is to find the best estimate $\mathbf{y}$ of the true target position $\mathbf{x}$. We assess filter performance for a single step and asymptotically (in a stable state). With the knowledge about target position at the previous step $\bar{\mathbf{x}}$, one-step performance is the filter performance at the next step (the current step). If the filters continue to be run over many steps, the estimate reaches a stable state where asymptotic performance can be derived.

Assuming a central unit with access to the local sensor data from all members in real time, CF simply stacks the local observations and uses a standard Kalman filter to track the target. Similarly, SF follows Kalman filtering and uses the sensor data at a single node only (without cooperation from the other node).

Let the error covariance of the position state $\bar{\mathbf{x}}$ be $\bar{\mathbf{P}}$. The estimate of the state $\bar{\mathbf{x}}$ is $\bar{\mathbf{y}} = \mathbb{E}[\bar{\mathbf{x}}]$. The one-step SF gives estimate with error covariance

$$\mathbf{P}_{\text{SF}} = \big(\mathbf{I} - (\bar{\mathbf{P}} + \mathbf{Q})(\bar{\mathbf{P}} + \mathbf{Q} + \mathbf{R}_1)^{-1}\big)(\bar{\mathbf{P}} + \mathbf{Q}), \tag{2}$$

and one-step CF gives

$$\mathbf{P}_{\text{CF}} = \big(\mathbf{I} - (\bar{\mathbf{P}} + \mathbf{Q})\mathbf{H}\mathbf{S}^{-1}\big)(\bar{\mathbf{P}} + \mathbf{Q}), \tag{3}$$

where

$$\begin{aligned} \mathbf{S} &= \mathbf{H}(\bar{\mathbf{P}} + \mathbf{Q})\mathbf{H}^\top + \begin{bmatrix} \mathbf{R}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_2 \end{bmatrix}, \\ \mathbf{H} &= \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix}, \end{aligned} \tag{4}$$

and $\mathbf{I}$ is the identity matrix of size $n \times n$.

When the target keeps moving and observations are made at every step at the two nodes with the same settings, filters reach stable state in which the estimation and performance approach constant values. The stable state estimation error covariances for both filters are derived by setting $\mathbf{P}_{\text{SF}} = \bar{\mathbf{P}}$ and $\mathbf{P}_{\text{CF}} = \bar{\mathbf{P}}$. In one dimension ($n = 1$), these reduce to:

$$\begin{aligned} \mathbf{P}_{\text{SF,ss}} &= \frac{-\mathbf{Q} + \sqrt{\mathbf{Q}^2 + 4\mathbf{Q}\mathbf{R}_1}}{2}, \\ \mathbf{P}_{\text{CF,ss}} &= \frac{-\mathbf{Q} + \sqrt{\frac{\mathbf{Q}^2(\mathbf{R}_1+\mathbf{R}_2)+4\mathbf{Q}\mathbf{R}_1\mathbf{R}_2}{\mathbf{R}_1+\mathbf{R}_2}}}{2}. \end{aligned} \tag{5}$$

The CF result is similar to SF, except that the observation fused uses the two sensor data. This 'fused' observation therefore has error covariance $(\mathbf{R}_1^{-1} + \mathbf{R}_2^{-1})^{-1}$.

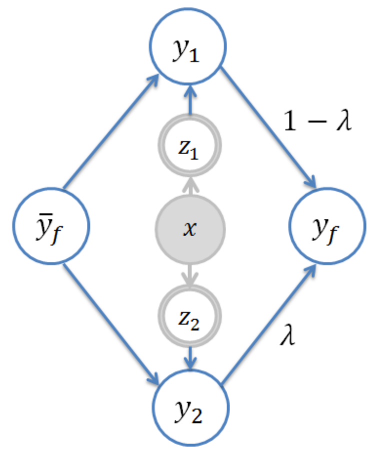



Fig. 2. Multi-sensor tracking: distributed filtering using weighted-sum fusion.

Do note that the estimates by CF and SF are consistent in that the actual error covariance of the estimate $\mathbb{E}[(\mathbf{y} - \mathbf{x})(\mathbf{y} - \mathbf{x})^\top]$ equals the estimated error covariance $\mathbf{P}$. Therefore we only state one of them here. The performance of CF and SF are shown in the subsequent sections.

## 3. Optimal Distributed Filtering

### 3.1. *Optimal DF*

The recursive two-step distributed filtering is shown in Fig. 2. At the previous step, sensor nodes exchange their local estimates and obtain a fused estimate $\bar{\mathbf{y}}_f$. This fused estimate of the previous position $\bar{\mathbf{x}}$ is adopted by both nodes. After that, the target position is predicted at each node locally, and updated with local sensor data using a standard Kalman filter. At the current step, sensor nodes exchange their local estimates $\mathbf{y}_1$ and $\mathbf{y}_2$ and obtain the fused estimate $\mathbf{y}_f$. We use a weighted sum to fuse the two estimates with weight $\lambda$:

$$\begin{aligned} \mathbf{y}_{f,\text{DF}} &= f(\mathbf{y}_1, \mathbf{y}_2) \\ &= (1 - \lambda)\mathbf{y}_1 + \lambda\mathbf{y}_2. \end{aligned} \tag{6}$$

Let the fused estimate at previous step $\bar{\mathbf{y}}_f = \bar{\mathbf{y}}$ with estimated error covariance $\bar{\mathbf{P}}_f = \bar{\mathbf{P}}$. Then the local estimates are for nodes $i = 1, 2$ are:

$$\begin{aligned} \mathbf{y}_i &= \mathbf{P}_i(\bar{\mathbf{P}}^{-1}\bar{\mathbf{y}} + \mathbf{R}_i^{-1}\mathbf{z}_i), \\ \mathbf{P}_i &= (\mathbf{R}_i^{-1} + \bar{\mathbf{P}}^{-1})^{-1}. \end{aligned} \tag{7}$$

The optimal weight $\lambda^*$ is obtained by minimizing the volume of the error covariance: $\det \mathbb{E}[(\mathbf{y}_{f,\text{DF}} - \mathbf{x})(\mathbf{y}_{f,\text{DF}} - \mathbf{x})^\top]$. In one dimension, we obtain:

$$\begin{aligned} \lambda^* &= \arg\min_{\lambda} \mathbb{E}[(\mathbf{y}_{f,\text{DF}} - \mathbf{x})^2] \\ &= \frac{\mathbf{R}_1}{\mathbf{R}_1 + \mathbf{R}_2}. \end{aligned} \tag{8}$$

This optimal fusion gives $(\mathbf{y}^*_{f,\text{DF}}, \mathbf{P}^*_{f,\text{DF}})$. This turns out to be identical with Bar Shalom's state vector fusion (SVF) [2] with

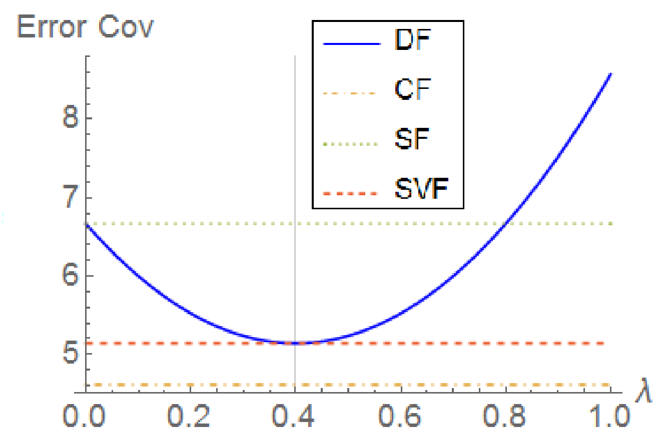



Fig. 3. One-step error covariances of DF, SF, CF and SVF, against the weight $\lambda$ used by DF ($P = 10$, $\mathbf{Q} = 10$, $\mathbf{R}_1 = 10$ and $\mathbf{R}_2 = 15$).

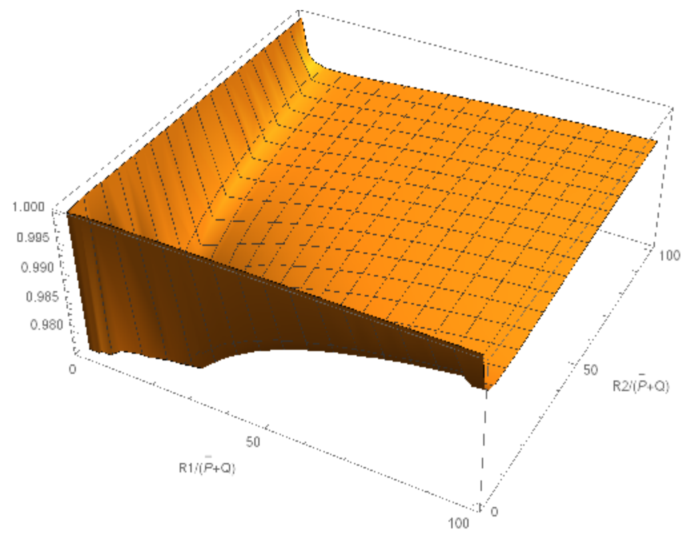



Fig. 4. Ratio of error covariances (optimal DF to CF) $\leq 1$.

different deviation approach:

$$\mathbf{y}_{f,\text{SVF}} = \mathbf{y}_1 + (\mathbf{P}_1 - \mathbf{P}_{12})(\mathbf{P}_1 + \mathbf{P}_2 - \mathbf{P}_{12} - \mathbf{P}_{12}^\top)^{-1}(\mathbf{y}_2 - \mathbf{y}_1),$$
$$\mathbf{P}_{f,\text{SVF}} = \mathbf{P}_1 - (\mathbf{P}_1 - \mathbf{P}_{12})(\mathbf{P}_1 + \mathbf{P}_2 - \mathbf{P}_{12} - \mathbf{P}_{12}^\top)^{-1}(\mathbf{P}_1 - \mathbf{P}_{12}^\top), \quad (9)$$

where the cross-correlation between the two estimates $\mathbf{P}_{12} = \mathbb{E}[(\mathbf{y}_1 - \mathbf{x})(\mathbf{y}_2 - \mathbf{x})^\top]$ is required. Although $\mathbf{P}_{12}$ is not required for optimal fusion in DF, it is required for a consistent estimate of the error covariance of the optimal fusion $\mathbf{P}^*_{f,\text{DF}}$. The stable state error covariance of optimal DF can also be derived by setting $\mathbf{P}^*_{f,\text{DF}} = \bar{\mathbf{P}}$.

### *3.2. Gap between the optimal DF and CF*

Fig. 3 shows an example of the one-step performance comparison of the error covariances of the estimates by DF, SF, CF and SVF, against the weight $\lambda$ used by DF. The optimal weight $\lambda^*$ is obtained at the lowest point of the DF curve, which gives identical performance as SVF. There is a gap between the optimal DF (or SVF) and CF. As stated by Bar-Shalom [6]:

> *The sufficient statistics for the global data set $D^{ij} = D^i \bigcup D^j$ cannot be expressed in terms of the sufficient statistics of the local data sets $D^i$ and $D^j$ (the local estimates $\hat{x}^i$ and $\hat{x}^j$).*

In other words, there is information loss by transmitting the processed data (estimates) instead of the raw data (observations). We explain the details next.

CF is a Maximum a Posteriori (MAP) estimator which estimates the unobserved target position state $\mathbf{x}$ with two observations $\mathbf{z}_1$ and $\mathbf{z}_2$. The prior distribution is known as $\mathcal{N}(\mathbf{x}, \bar{\mathbf{P}} + \mathbf{Q})$. On the other hand, the distributed filter treats the two local estimates $\mathbf{y}_1$ and $\mathbf{y}_2$ as two 'observations'. The fusion is a Maximum Likelihood (ML) estimation without using the 'prior' about the state [5]. This is the best that distributed estimation can achieve when only local estimates are available for fusion. Therefore it is optimal only in the ML sense. We take SVF as example; it is identical to the optimal DF. Let the error covariance of the two estimates from the nodes be

$$\mathbf{P}_{\text{SVF}} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_{12} \\ \mathbf{P}_{12}^\top & \mathbf{P}_2 \end{bmatrix}, \quad (10)$$

where

$$\begin{aligned} \mathbf{P}_1 &= \mathbb{E}[(\mathbf{y}_1 - \mathbf{x})(\mathbf{y}_1 - \mathbf{x})^\top], \\ \mathbf{P}_2 &= \mathbb{E}[(\mathbf{y}_2 - \mathbf{x})(\mathbf{y}_2 - \mathbf{x})^\top], \\ \mathbf{P}_{12} &= \mathbb{E}[(\mathbf{y}_1 - \mathbf{x})(\mathbf{y}_2 - \mathbf{x})^\top]. \end{aligned} \quad (11)$$

The target position $\mathbf{x}$ is to be determined based on the two 'observations' $\mathbf{y}_1$ and $\mathbf{y}_2$. The logarithm of the likelihood function $\mathcal{L}(\mathbf{x}; \mathbf{y}_1, \mathbf{y}_2)$ is

$$\begin{aligned} &\ln \mathcal{L}(\mathbf{x}; \mathbf{y}_1, \mathbf{y}_2) \\ &= \ln 2\pi - \frac{1}{2} \ln \Big( \det \mathbf{P}_{\text{SVF}} \\ &+ \left( \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{x} \\ \mathbf{x} \end{bmatrix} \right)^\top \mathbf{P}_{\text{SVF}}^{-1} \left( \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{x} \\ \mathbf{x} \end{bmatrix} \right) \Big). \end{aligned} \quad (12)$$

It is maximized by setting $\frac{\partial \mathcal{L}(\mathbf{x}; \mathbf{y}_1, \mathbf{y}_2)}{\partial \mathbf{x}} = 0$. The optimal estimate for position state $\mathbf{x}$ in one dimension is

$$\begin{aligned} \mathbf{y}^*_{\text{ML}} &= \arg\max_{\mathbf{x}} \mathcal{L}(\mathbf{x}; \mathbf{y}_1, \mathbf{y}_2) \\ &= \frac{\mathbf{P}_2 \mathbf{y}_1 + \mathbf{P}_1 \mathbf{y}_2 - \mathbf{P}_{12}(\mathbf{y}_1 + \mathbf{y}_2)}{\mathbf{P}_1 + \mathbf{P}_2 - 2\mathbf{P}_{12}} \end{aligned} \quad (13)$$

This is the same as the fused estimate by SVF and the optimal DF. We compare the error covariance with the one by CF using:

$$\frac{\det \mathbf{P}_{f,\text{SVF}}}{\det \mathbf{P}_{\text{CF}}} = \frac{\det \mathbf{P}^*_{f,\text{DF}}}{\det \mathbf{P}_{\text{CF}}}. \quad (14)$$

Fig. 4 shows that the ratio is always less than one for $n = 1$. The axes are $\mathbf{R}_1$ and $\mathbf{R}_2$ normalized by $\bar{\mathbf{P}} + \mathbf{Q}$.

## 4. Naïve Filtering

The NF simply fuses the two estimates assuming no correlation between them. The procedure is similar to DF in Fig. 2

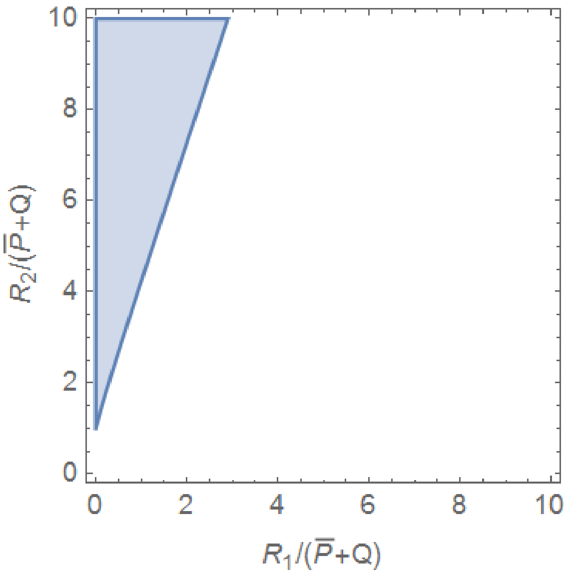



Fig. 5. One-step performance: the dangerous region of implementing NF (assuming $\mathbf{R}_1 < \mathbf{R}_2$).

but the weight of fusion is calculated by the estimated error covariances, that is, $\lambda_{\text{NF}} = \mathbf{P}_{f,\text{NF}}\mathbf{P}_2^{-1}$. The fused estimate is:

$$\begin{aligned} \mathbf{y}_{f,\text{NF}} &= (\mathbf{I} - \lambda_{\text{NF}})\mathbf{y}_1 + \lambda_{\text{NF}}\mathbf{y}_2, \\ \mathbf{P}_{f,\text{NF}} &= (\mathbf{P}_1^{-1} + \mathbf{P}_2^{-1})^{-1}, \end{aligned} \tag{15}$$

where $\mathbf{P_1}$, $\mathbf{P_2}$, $\mathbf{y_1}$ and $\mathbf{y_2}$ are local estimates from (7). The resulting error covariance of the fused estimate is $\mathbb{E}[(\mathbf{y}_{f,\text{NF}} - \mathbf{x})(\mathbf{y}_{f,\text{NF}} - \mathbf{x})^\top]$.

In NF, the common information from the two estimates is double-counted in the fusion; this leads to estimation overconfidence. The estimated error covariance $\mathbf{P}_{f,\text{NF}}$ is smaller than the actual error covariance, i.e.,

$$\det \mathbf{P}_{f,\text{NF}} < \det \mathbb{E}[(\mathbf{y}_{f,\text{NF}} - \mathbf{x})(\mathbf{y}_{f,\text{NF}} - \mathbf{x})^\top]. \tag{16}$$

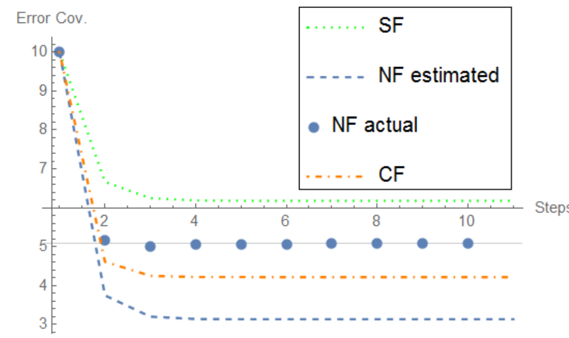
In fact, it is even smaller than the one by CF! The overconfidence prevents utilization of subsequent useful information and therefore the actual error covariance of NF estimate sometimes is even worse than SF without cooperation. In such case, cooperation using NF is no longer advantageous. We derive all the cases when cooperation using NF is worse than SF. The region derived is called the *dangerous region* where the naïve assumption completely fails:

$$\det \mathbb{E}[(\mathbf{y}_{f,\text{NF}} - \mathbf{x})(\mathbf{y}_{f,\text{NF}} - \mathbf{x})^\top] > \det \mathbf{P}_{\text{SF}}. \tag{17}$$

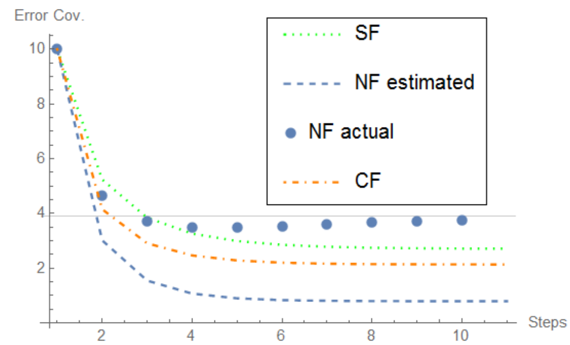
### 4.1. One-step performance

We calculate the one-step dangerous region when the inequality in (17) is met. For $n = 1$, without loss of generality, we assume $\mathbf{R}_1 \leq \mathbf{R}_2$. The dangerous region is obtained as:

$$\begin{aligned} \mathbf{r}_2 &> 1, \\ \mathbf{r}_1 + \mathbf{r}_2 + 3\mathbf{r}_1\mathbf{r}_2 &\leq \mathbf{r}_2^2, \end{aligned} \tag{18}$$




(a) $\bar{P} = 10$, $\mathbf{Q} = 10$, $\mathbf{R}_1 = 10$ and $\mathbf{R}_2 = 15$




(b) ($\bar{P} = 10$, $\mathbf{Q} = 1$, $\mathbf{R}_1 = 10$ and $\mathbf{R}_2 = 20$)

Fig. 6. Naïve filter in stable state: (a) Actual error covariance is smaller than SF. (b) Actual error covariance eventually gets worse than SF (dangerous case). In both cases, NF is overconfident about its estimation. The estimated error covariance by NF is even lower than the one by CF.

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are the normalized error covariances and $\mathbf{r}_i = \mathbf{R}_i/(\bar{\mathbf{P}} + \mathbf{Q}), i = 1, 2$. This is plotted in Fig. 5. We can see that one-step NF is only safe to use if both local observations have very small errors compared with the propagation error, or when the two local sensors have comparable observation errors.

### 4.2. Asymptotic performance

After the first step, if we keep using the naïve assumption and fuse the estimates with weights calculated from the estimated error covariances, the actual estimation error could diverge and become even worse. Fig. 6 shows two cases in stable state. The blue dots are the actual (by Monte Carlo simulation) error covariances of the NF estimates. The horizontal line is the calculated asymptotic error covariance for NF estimates. The blue dashed line shows the estimated error covariances given by NF. We can see that NF is overconfident about its estimates. In case (b), even though in the first step (step 2) NF gives improvement over SF, its estimation error becomes worse than SF later.

We calculate the stable state error covariance of NF estimates. Compared with stable state SF error covariance, the asymptotic dangerous region is plotted in Fig. 7. Note that the asymptotic performance does not depend on initial error covariance, and therefore the axes are $\mathbf{R}_1$ and $\mathbf{R}_2$ normalized

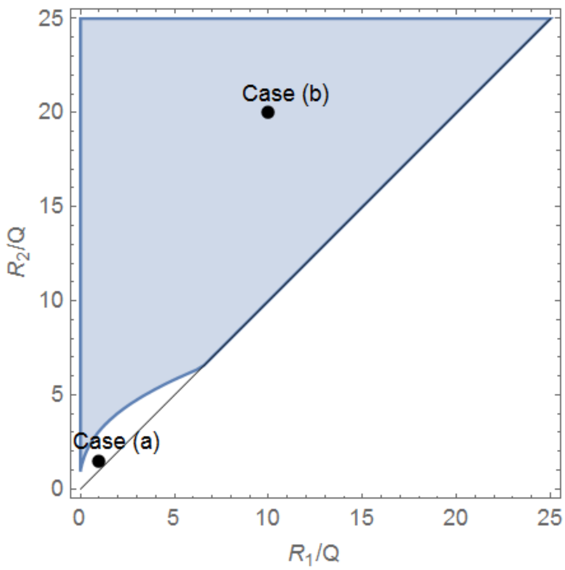



Fig. 7. Asymptotic performance: the dangerous region of implementing NF (assuming $\mathbf{R}_1 \leq \mathbf{R}_2$). Cases (a) and (b) from Fig. 6 are located.

by the process noise error covariance $\mathbf{Q}$. The two cases in Fig. 6 are located in the dangerous region plot; case (b) falls into the dangerous region. We can see that unless the two local sensors have small and comparable measurement errors, it is dangerous to use naïve filter for cooperation.

## 5. Conclusion and Future Work

We examined the performance of distributed processing in an underwater multi-sensor tracking problem. We showed that the optimal distributed filter is achieved only when inter-dependency is tracked. However, there is still information loss due to transmission of processed data instead of the raw data. We also showed the consequence of implementing Naïve filtering is estimation overconfidence. The actual estimation could be worse than the single filter without cooperation. We derived the dangerous regions for naïve filter operation. This can be used as a guideline for using the naïve assumption for underwater cooperative localization.

The multi-vehicle localization problem is related, but nodes in the team estimate their own positions, and cooperation happens with an additional relative measurement relating the two states. We are currently working to answer similar questions as in this paper, but in the context of underwater multi-vehicle localization.

## References


[1] Julier, SJ and Uhlmann, JK, General Decentralized Data Fusion With Covariance Intersection (CI). In: Hall, D and Llinas, J, (eds.) Handbook of Data Fusion. CRC Press: Boca Raton FL, USA (2001)

[2] Y. Bar-Shalom and L. Campo, "The effect of the common process noise on the two-sensor fused-track covariance," Aerospace and Electronic Systems, IEEE Transactions on, vol. AES-22, pp. 803-805, Nov 1986.

[3] Yaakov Bar-Shalom and Xiao-Rong Li, Multitarget-Multisensor Tracking:Principles and Techniques, 1995

[4] J. A. Roecker and C. D. McGillem, "Comparison of two-sensor tracking methods based on state vector fusion and measurement fusion," in IEEE Transactions on Aerospace and Electronic Systems, vol. 24, no. 4, pp. 447-449, Jul 1988.

[5] K. C. Chang, R. K. Saha and Y. Bar-Shalom, "On optimal track-to-track fusion," in IEEE Transactions on Aerospace and Electronic Systems, vol. 33, no. 4, pp. 1271-1276, Oct. 1997.

[6] Y. Bar-Shalom, "Comments on "Comparison of two-sensor tracking methods based on state vector fusion and measurement fusion" by J. Roecker et al," in IEEE Transactions on Aerospace and Electronic Systems, vol. 24, no. 4, pp. 456-457, July 1988.