A Simulation Framework for Decentralized Data Fusion Networks

Viktor Deleskog
Department for C4ISR, Sensor Informatics
FOI, Swedish Defence Research Agency
SE-164 90 Stockholm
SWEDEN

viktor.deleskog@foi.se

ABSTRACT

This paper presents software framework for data fusion networks in dynamic target tracking applications. The idea behind the framework is to facilitate implementation and evaluation of complete decentralized fusion networks or specific parts, e.g. strategies for communication and data compression. The framework consists of classes that abstract essential building blocks required in a decentralized data fusion network. The framework provides a tool for fast prototyping and facilitate evaluation of decentralized fusion networks in target tracking applications. The research interest in multi-static and mobile sensors, communication and fusion networks indicates the importance of such a tool. Specifically to provide data for assessment of expected performance with respect to different choice of fusion architectures, algorithms and sensor types.

1.0 INTRODUCTION

Data fusion in a network consisting of multiple and heterogeneous sensors is an important area of research, specifically target tracking for situational awareness. Increasing number of sensors and abilities with mobile sensors raise issues of what data to communicate and how to do it. In an optimal setup, all sensors can communicate with a central fusion center. This is not always the case due to limited communication bandwidth and it is probable not the best solution for large networks of sensors. Decentralized fusion architectures offers a more flexible and robust way of fusing data since some level of fusion can be performed locally by the sensor itself, as a local fusion node. The locally fused data is a more compact and refined representation of the sensor data that can be communicated with other nodes in a network. In addition, this enables integration of sensors that only expose fused data instead of raw data.

In target tracking, fused data corresponds to estimated kinematic tracks of detected targets. In a decentralized network for target tracking estimated tracks are communicated and fused instead of raw detections from *e.g.* a radar or a camera sensor. To fuse tracks *track-to-track fusion* (T2TF) methods are applied. Methods for exact T2TF exists [1]. The drawback of exact methods is that full knowledge about the correlations between estimates are required, which in turn requires much more information to be communicated and it is probably not available.

To remedy this, one can apply fusion algorithms that handle unknown correlations, such as *inverse covariance intersection* (ICI, [2]). The ICI algorithm shares the same approach as with the well known *covariance intersection* (CI, [3]) algorithm. The algorithm handles existence of unknown covariance between the estimates to be fused which is central in decentralized fusion networks. The ICI algorithm is a less conservative alternative than CI but still consistent, in many cases. The algorithm has been tailored for situations where common information between estimates are existent. This is generally the case in decentralized target tracking networks where a common target model is used to track detected targets. Previous work is presented in [4] and [5] where real data is used to evaluate such fusion algorithms in a surveillance application.

In the information domain, fusion infrastructures such as c*hannel cache* (CC, [6]) can be applied to handle that information is not reused, i.e. double counted. The channel cache is based on a single connected tree topology network inspired by the *channel filter* [7]. The channel cache, compared to channel filter does not keep track of the common information between nodes when transmitting. This implies an easier way to communicate since a node just transmits its local independent information. The received information is stored in a local *channel cache* that maintains the history of contributed information for each track and node. Since information is additive, delayed information can easily be added in the cache, *e.g.* on temporarily communication loss. From an integration point of view it makes it possible for ad-hoc connectivity of sensors, such as an UAV that is activated to update a specific target. At a node, a global track is built by just summing all local and cached received information from a given target.

To achieve a common situational picture in the network, effective communication is needed. Especially when links are limited in bandwidth and communication conditions varies. Sending data over limited communication links requires strategies for what to send and when. Handling of dynamic communication conditions requires that the network topology is robust to sporadic loss of single nodes.

These aspects motivate a framework to facilitate implementation and simulation of decentralized fusion networks. Here we focus on decentralized target tracking applications and how different choice of fusion methods, data association and communication strategies affect tracking performance. A software framework written in MATLAB™ based on an abstract class architecture along with tools for simulation and evaluation is presented (Section 2.0). At last, the framework is used in an example case study to present how it can be used in a particular area surveillance scenario to evaluate different choices of communication strategies and fusion infrastructures (Section 3.0).

2.0 SIMULATION FRAMEWORK

The framework is divided in two parts: *architecture* and *simulation*. The first part consists of building blocks for implementation and design that abstract the core functionalities of a decentralized fusion network. The second part handles simulation to support evaluation of implemented networks in defined scenarios.

2.1 Architecture

The most fundamental building blocks in the architecture are nodes and links. A *node* is an abstraction of an object that can connect with other nodes to create a network. A *link* is an abstraction of the communication channel between two nodes and the data transportation. A node which fuse data is called a *fusion node.* The fusion node inherits the node class with additional properties and functionality. It is still abstract and must be implemented by the user. Required functionality is represented by objects with a common interface to facilitate implementation. These objects abstract local fusion, track fusion, data association and communication management. Node, link and fusion node objects are grouped within *network and fusion objects*. To connect these objects with physical objects we have the objects: *platform* and *sensor*. A *platform* object abstracts a physical object with attributes such as position and velocity. The *sensor* object abstracts a physical sensor that can observe the physical world and generate observations according to a measurement model. The described objects above, illustrated in Figure 1, are the core objects that forms the fundamentals of the implementation part of the framework.

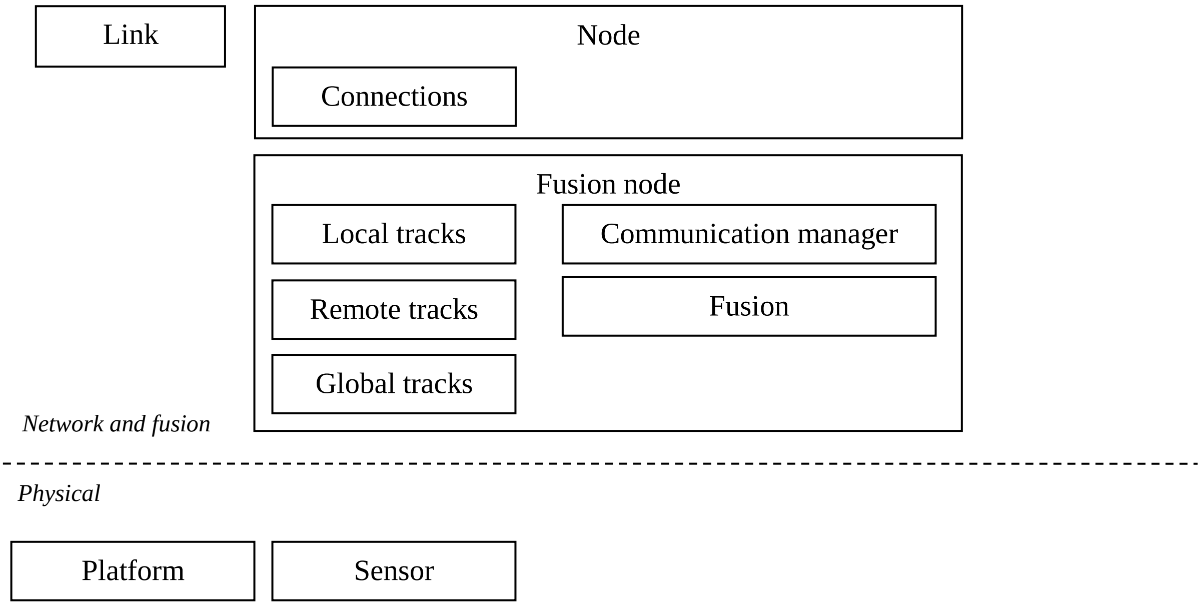

Figure 1: The architecture of the framework where boxes represents different objects. Base objects are link, node, platform and sensor. The fusion node inherits the node object with additional properties and functionality. The dashed line separates network and fusion objects from physical objects.

To describe the fusion node in more detail it is illustrated in Figure 2 with its properties, functionalities and data flow. The arrows show the flow of data between internal objects and external objects. Abstract functionality objects are represented with thick borders. Using measurements from mounted sensors, tracks are estimated by the local fuser object, *e.g.* a target tracker, and stored as local tracks. The communication manager encapsulates all functionality to handle the logics of transmission and reception of data. For example, a smart logic that compares local and received data and just transmit the most informative data can be implemented in the communication manager. Received track data is stored as remote tracks. Generation of global tracks is handled by the *track fuser*, using associations computed by the *associator* object. Here, an association represent the hypothesis that a local or remote track originates from the same target as a global track. Using this architecture makes it possible to implement and design varies types of fusion infrastructures and combine different fusion algorithms and association logics.

2.2 Simulation

The network and fusion objects support simulation by applying a discrete time stepping scheme. Simulation runs in a time loop where objects are stepped forward in time in a specific order. Physical objects are integrated in the simulation by common interfaces to support either random generation of data or reading predefined data, *e.g.* trajectories and detections. When the simulation loop starts, data for physical objects are generated or read. During the simulation three steps are repeated for each time step: advance physical objects, process available new data in all nodes and advance links. The simulation loop can be extended to handle multiple simulations, *i.e.* Monte Carlo.

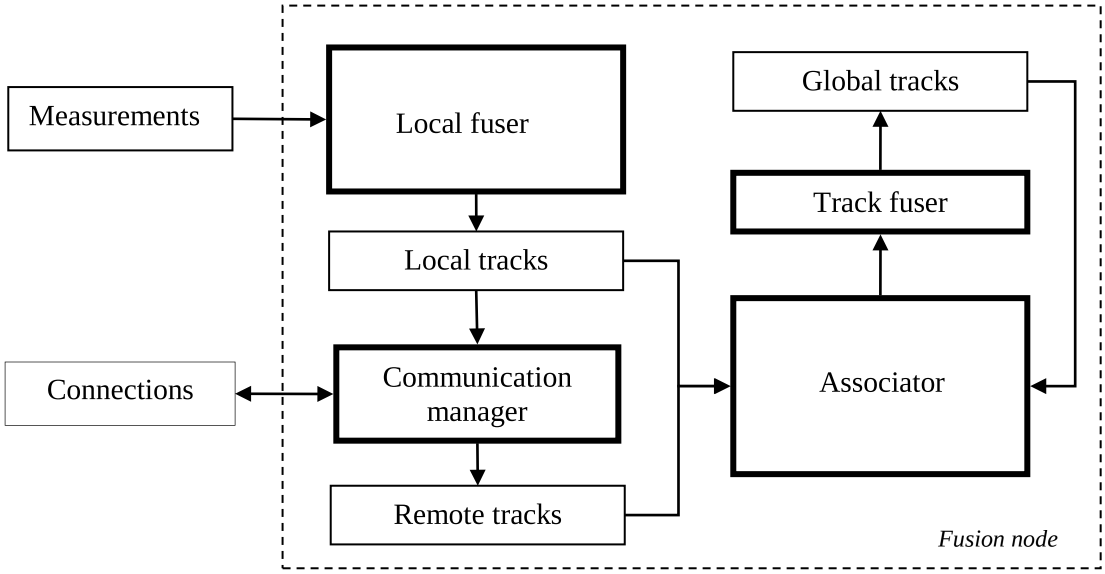

Figure 2: The fusion node object in more detail. The dashed line represent the boundary of the fusion node. Boxes with bold lines are abstract objects to be implemented by the user. Local measurements are processed by the local fuser. The communication manager handles how data is transmitted and received. Global tracks are generated by the track fuser using associations computed by the associator.

3.0 CASE STUDY: REDUCED COMMUNICATION

To show how the framework can be applied, it will used in an example case study. An area surveillance application is studied consisting of a combination of two stationary camera sensors and one mobile sensor. The mobile sensor is in this case an UAV equipped with a looking down camera sensor. The cameras are modelled as radar sensors with a limited field of view that are measuring azimuth and elevation angles to a target. Each sensor is acting as a fusion node that communicates local estimated tracks. Local tracking in each fusion node is performed by a *multi-hypothesis tracking* (MHT, [8]) tracker in single hypothesis mode, where each track is maintained by an e*xtended Kalman filter* (EKF) combined with a *constant velocity* (CV) target model in 3-D. Track-to-track associations are computed using the standard auction algorithm [8] during a time window to incorporate target dynamics. The *channel cache* and *inverse covariance intersection* (ICI) fusion methods are chosen, since their difference approach on how data is handled and fused.

To evaluate performance we apply the *generalized optimal sub-pattern assignment* (GOSPA, [9]) metric, which is suitable in target tracking contexts. GOSPA is computed with $p = 2$ and $\alpha = 2$ and cut-off distance 20 m. Centralized tracking serves as a baseline to compare how different configurations perform against the "optimal" configuration, where all measurements are available at a central fusion center. Evaluation data is generated from 100 Monte Carlo runs where the means over all simulations are used. The sampling time is one second.

3.1 Case Study: Reduced communication

In this study, reduced communication is evaluated on how it affects the tracking performance with respect to fusion method, in this case channel cache and ICI. Reduction is performed by skipping a number of state updates and only transmit the latest update in each transmission. All nodes communicate with each other over ideal point-to-point links. A single target is moving on a plane with constant speed from the right to the

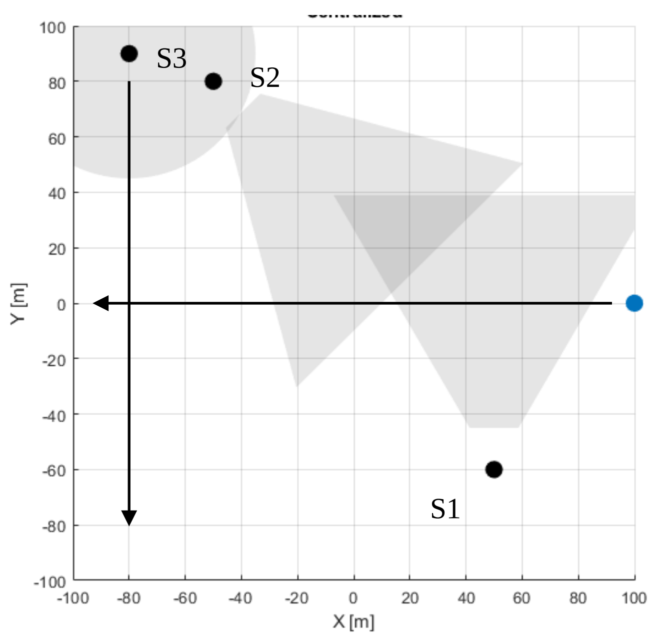

Figure 3: Scenario used in the example study. A target (blue circle) moves to left and is observed by three sensors: S1, S2 and S3. S1 and S2 are stationary sensors while S3 is an UAV moving downwards.

left at 1 *m/s*. During its movement it will be detected and tracked by S1 first and then by S2. S1 and S2 are placed above the plane and directed slight downwards. When the target exists the field of view of S2 it will be detected and tracked by S3. S3 is flying in the negative Y-axis direction with a speed of 1 *m/s* and the sensor is directed straight downwards on the plane, see Figure 3. Sensors are illustrated as black circles and the target with a blue circle. The sensors field of view are projected on the plane as grey patches. The arrows illustrate the velocity vector of the target and S3. The scenario covers the practical usefulness of how a moving sensor can assist a stationary network of sensors with limited area coverage. Each sensor generates a measurement of the target with probability 0.7, if the target is in its field-of view, otherwise 0. The scenario runs for 95 seconds.

Table 1 gives a summary of mean GOSPA for all nodes including fusion method and three communication reduction configurations: C0, C2 and C5. C0 is full communication, *i.e.* no reduction, C2 transmits every other second and C5 transmits every fifth second. In the first column we see that local tracking is a factor of three greater than the baseline. This is since local tracking is penalized for not detecting the existing target at all. When information is communicated between the nodes we see that both channel cache and ICI achieves almost same performance. In the ideal communication configuration, *i.e.* C0, we see that they are better than the baseline configuration. The measurement model is non-linear so it is possible that non-linear effects can influence the computations and eventually accumulated in the communicated terms. Still, channel cache performs better than ICI for all reduction configurations. ICI is an approximation of exact fusion which may explain that all information is not exploited in the fused estimate. On the other hand, ICI do not need full bookkeeping as channel cache to prevent double counting of information.

In Figure 4, the underlying GOSPA data in Table 1 is displayed with respect to time to give a better illustration of achieved tracking performance. Fusion methods and nodes are divided by column and row respectively. The baseline is illustrated with a thick grey line and local tracking by solid black line. Configuration C0 is represented by dashed line, C2 by dash-dotted line and C5 by dotted line. Firstly, we see

Table 1 Computed mean GOSPA for nodes, fusion method and communication reduction configurations: C0, C2 and C5. First column is local tracking and baseline is centralized tracking, i.e. the "optimal" configuration. When communication is reduced up to 5 seconds some degradation is present for both channel cache and ICI.

		Channel cache			ICI		
	Local	C0	C2	C5	C0	C2	C5
Node 1	9.30	3.69	3.76	4.68	3.80	4.11	4.80
Node 2	10.81	3.69	3.82	4.86	3.80	4.16	5.08
Node 3	10.90	3.69	3.84	4.77	3.78	3.82	4.78
Baseline	3.84						

successful takeover by the nodes and how the global situation picture at each node keep up with the baseline. No significant difference between the fusion methods are visible for C0 and C2. With configuration C5 some difference are visible, especially for S3 when it just relies on received information from S1 and S2. In the end, the performance for S1 and S2 with C5 degrades as they only relies on prediction while waiting for new information from S3. To conclude, no significant performance difference is noticeable between channel cache and ICI. Both keep up with the centralized setup. Performance degradation is existent when the communication interval is increased, which is intuitive.

4.0 CONCLUSION

This paper has presented a simulation framework for decentralized data fusion networks. The basic architecture and its fundamental building blocks are described. The framework enables implementation of a whole fusion network or its specific parts. Using the framework facilitates evaluation of fusion performance using simulation of sensors, platforms and data communication. An illustrative example study is performed to show how the framework can be used to evaluate decentralized target tracking in a reduced communication setting.

Mainly, the framework is a tool that enable us to study how a decentralized fusion ability contributes to a more robust and effective use of available sensors, both stationary and mobile. Further, which fusion architecture is best suited in configurations where just target estimates are available from some sensors in combination with sensors that offer full access to raw measurements. Currently, the framework is under continuous development and has been used in research projects that study autonomous area surveillance and autonomous reconnaissance with swarming UAVs.

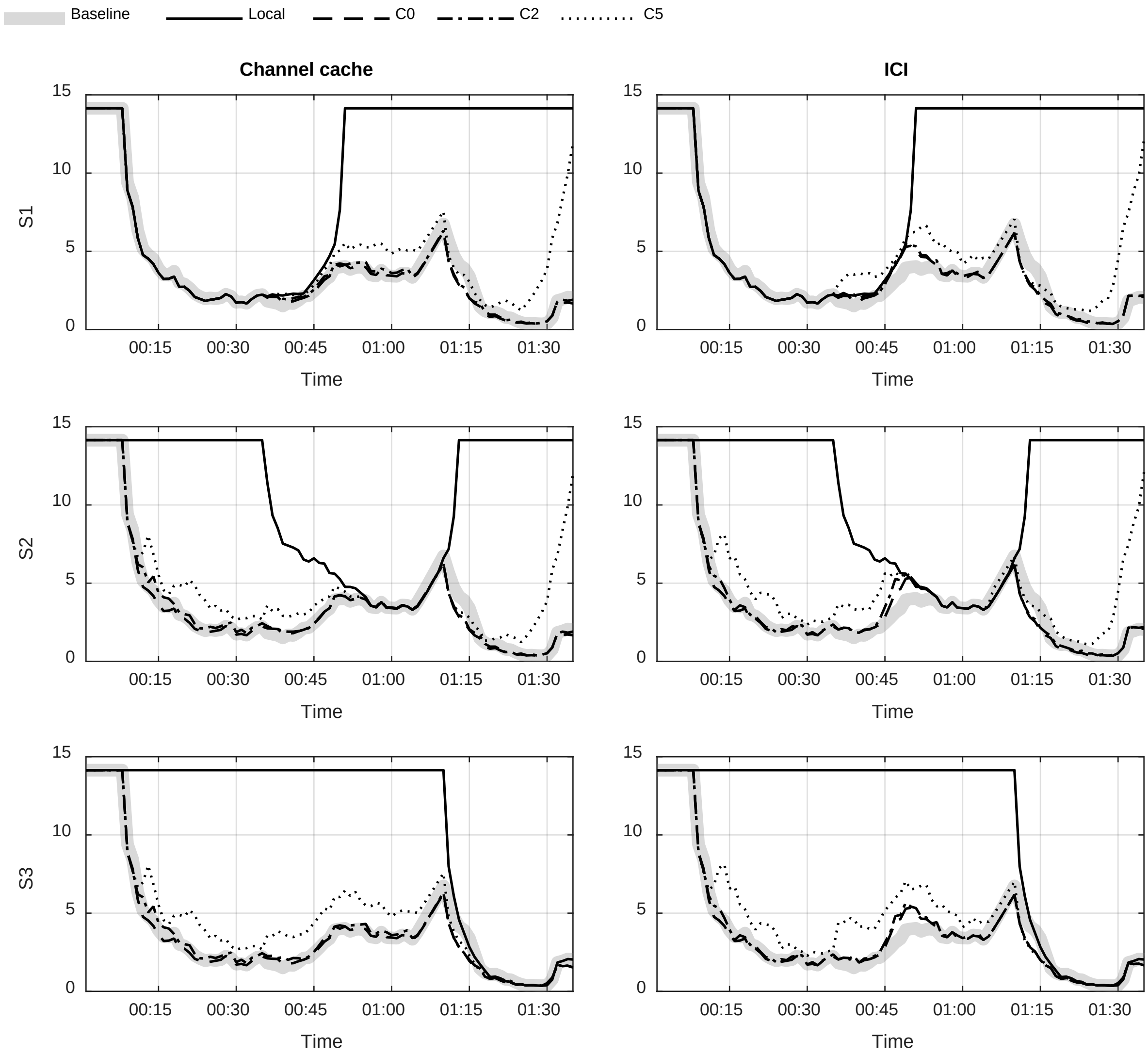

Figure 4: GOSPA with respect to time for nodes (rows) and fusion method (columns). Thick grey line represent the baseline, solid line local tracking at each node and communication reduction configurations are illustrated with dashed, dashed-dotted and dotted lines.

5.0 REFERENCES

[1] X. Tian and Y. Bar-Shalom. Exact algorithms for four track-to-track fusion configurations: All you wanted to know but were afraid to ask. In *Proc. 12th IEEE Int. Conf. Inform. Fusion*, pages 537–544, 2009.

[2] B. Noak, J. Sijs, and U. D. Hanebeck. Inverse covariance intersection: New insights and properties. In *Proc. 20th IEEE Int. Conf. Inform. Fusion*, Xi'an, China, July 10–13 2017.

[3] S. J. Julier and J. K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proc. American Control Conference*, Albuquerque, NM, USA, June 1997.

[4] J. Nygårds, V. Deleskog, and G. Hendeby. Safe fusion compared to established distributed fusion methods. In *Proc. IEEE Int. Conf. on Multisensor Fusion*, Baden-Baden, Germany, Sept. 2016.

[5] J. Nygårds, V. Deleskog and G. Hendeby. Decentralized Tracking in Sensor Networks with Varying Coverage. In *Proc. 21st IEEE Int. Conf. Inform. Fusion*, Cambridge, United Kingdom, July 2018.

[6] D. Hall, Chee-Yee Chong, J. Llinas, and M. Liggins. Distributed Data Fusion for Network-Centric Operations. *CRC Press, Inc.*, 2013

[7] K. Chang, Chee-Yee Chong and S. Mori. On scalable distributed sensor fusion. In *Proc. 11th Int. Conf. Inform. Fusion*, Cologne, Germany, July 2008.

[8] S. Blackman and R. Popoli. Modern tracking systems. *Artech House*, 1999.

[9] A. S. Rahmathullah, Á. F. García-Fernández and L. Svensson. Generalized optimal sub-pattern assignment metric. In *Proc. 20th IEEE Int. Conf. Inform. Fusion*, Xi'an, China, July 10–13 2017.