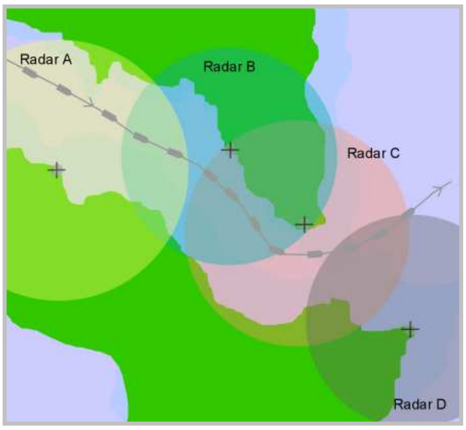

SPx-FUSE

Radar Track Fusion

Technical Overview

Issue 1.3
9th August 07

Table Of Contents

1 Introduction

SPx-FUSE, a component of the SPx radar processing family, is a COTS software module that combines plot and track information from multiple trackers into a single correlated track. For a complex radar installation with multiple overlapping radar, SPx-FUSE provides a single integrated track database by combining tracks from the multiple sources. The module can be set-up from configuration files to allow turn-key operation and can be monitored and controlled by a user interface running either locally or remotely to the system.

SPx-FUSE can be configured to receive radar plot and track reports from up to 8 sources. Positional information for each remote source is configured in SPx-FUSE and retained in configuration files which are loaded on start-up of the module. During operation, SPx-FUSE receives track reports from each sensor. There is considerable flexibility in the data that can be processed by SPx-FUSE, with the standard set of track descriptions including filtered and measured position, speed and size. SPx-FUSE associates observations from one sensor with any overlapping sensors and identifies measurements that are believed to derive from the same target.

The combination rules for processing multiple reports allow for flexibility in the configuration of the system. In one mode, the total coverage area can be mosaiced, so that a report from a preferred radar is used at each location of the target. In this mode, SPx-FUSE selects the correct track report as a function of the target's position, switching sources as the target moves across the boundary of the mosaic. In another mode, SPx-FUSE selects the best track on the basis of a quality measure, which is can either be calculated by the module automatically, or else derived from information in the incoming track report.

An internal track database retains the current track information which is periodically sent to remote clients over an Ethernet network. Standard TCP/UDP protocols are used to facilitate distribution to multiple clients in a multicast mode. The same network may be used to control and monitor SPx-FUSE with a graphical user interface that provides an animation of the internal state. This can be used to verify the receipt of data from the remote trackers and ensure valid data is being delivered from SPx-FUSE.

2 SPx-FUSE Architecture

SPx-FUSE operates as either a self-contained application running under Windows or Linux, or else as a software library that you build into your application. The full capabilities of SPX-FUSE are available in both situations.

2.1 SPX-FUSE as a standalone application

As a standalone application, SPX-FUSE starts as a process or service and initialises itself using data in configuration files. The module presents a network interface to a controlling application, which allows the application to send commands to SPX-FUSE, as well as receive track reports and status information. SPx-FUSe may be fully initialised and controlled over the network interface.

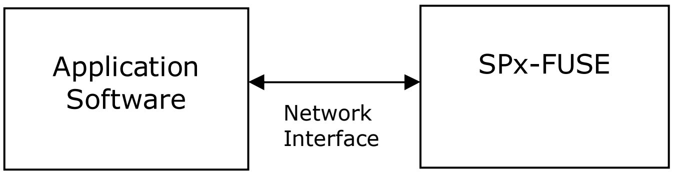

Figure 1 - SPx-FUSE as standalone application

2.2 SPX-FUSE as a software library

As a software library, all the capabilities of SPX-FUSE are available to your application, which accesses the functions through well-defined C/C++ interfaces. The software interface provides the means to:

- Tell SPx-FUSE the format and location of track reports to expect
- Tell SPx-FUSE how to combine track reports
- Tell SPx-FUSE how and where to output correlated reports
- Dynamically change parameters associated with the fusion process

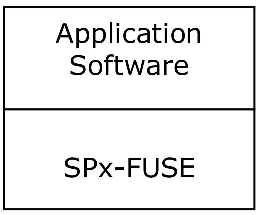

Figure 2 - SPxFUSE as software library

3 SPx-FUSE Functions

SPx-FUSE accepts track or plot reports from a number of remote trackers and combines the data to output reports of improved quality.

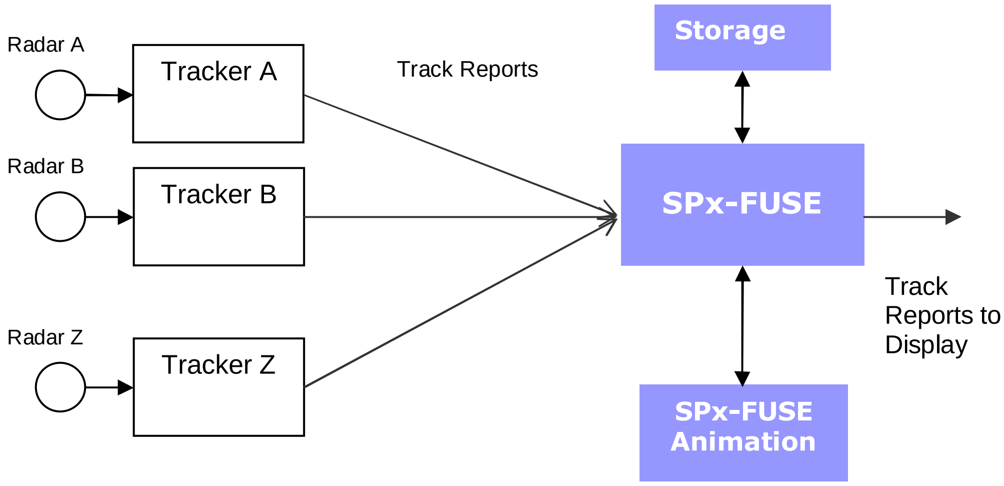

Figure 3 - SPxFUSE configuration

SPx-FUSE has a number of applications, including:

- Automatic track correlation from overlapping sensors to select the "best" report for onward delivery.
- It can maintain track identities as a target passes from one tracker to another.
- It can record tracks for later replay
- It can translate tracks from one format (input) to another (output).
- Provides a display of track and fusion activity (Animation)

3.1 *Plot and Track Fusion*

SPx-FUSE is able to process both filtered (track) and measured (plot) target positions. In the case of plot fusion, the remote tracker is assumed to be tracking the target and sending measured plot positions to SPx-FUSE. Significantly, there needs to be a remote tracker, not just a plot extractor.

3.2 Overlapping Sensors

In the simple situation of Figure 4, a target is moving from the radar coverage of Radar 1 through to the radar coverage of Radar 2. The area of coverage of the two radars is indicated by a circle. Initially, the target is detected by Radar 1 and maintained as a track by the local tracker associated with that radar. Eventually, the target moves into the region that is the overlap between Radar's 1 and 2. At this point, the local track associated with Radar 2 picks up the target and there are track reports generated for the same target by both trackers.

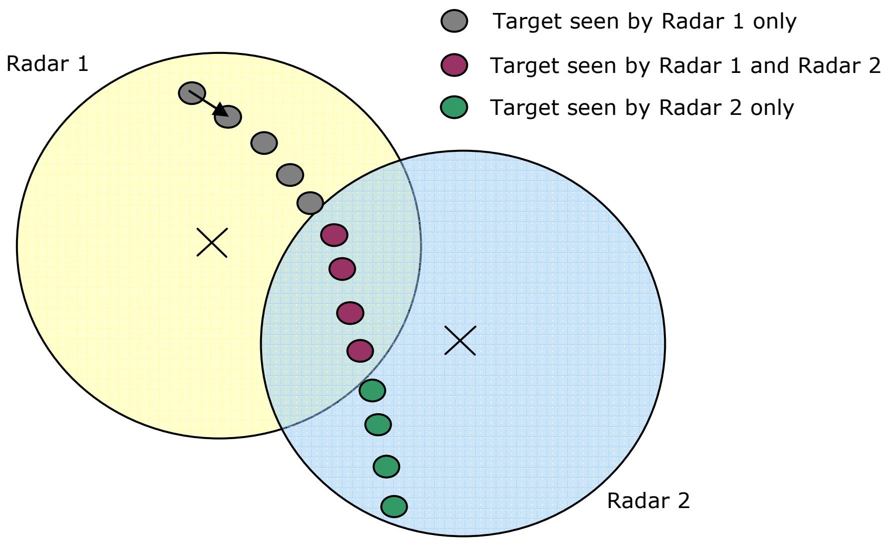

Figure 4 - Overlapping Sensors

The role of track fusion is to combine the track reports from the two local trackers into a single report that provides the "best" single viewport. In the context of the above example, the SPx-FUSE module provides the following key functions:

- It receives track reports from the local tracks associated with radars 1 and 2.
- It identifies whether that a target is covered by two radars and combines track reports.
- It preserves track identifiers as the target moves from Radar 1 to Radar 2.

For the first of these, each radar is assumed to be connected to a tracker and the output of the tracker is the input to SPx-FUSE. Note that the tracker may be one supplied by Cambridge Pixel, ie SPx-TRACK, or may be another type that SPx-FUSE can interface to.

In determining whether track reports from the two trackers may be reporting the same target, SPx-FUSE uses the geometry of the radar centres and ranges to compute the region of intersection. Track reports from the two trackers are then subject to a test of association to see if they are likely to be two observations of the same target. If this is the case, the track reports are combined (of course taking into account that the individual track reports are derived at different times, so will not in general be observing the target in the same position).

3.3 Preserving Track Identifiers

SPx-FUSE can preserve track identifiers across multiple trackers. Each local tracker will, in general, allocate its own track id to a newly acquired target. As the target moves from one radar coverage to another it is desirable to preserve the track Id. SPx-FUSE handles this by allocating its own track identifier at the point where the track from Radar 1 is first observed. Then as the target moves into the coverage of Radar 2, the id allocated from Radar 2 local tracker is converted into the SPX id. More details about the generation of track identifiers is given later in this document.

4 SPx-FUSE Processing

4.1 *Defining the Radar Coverage*

SPx-FUSE needs to understand the geometry of the radars, notably their position and coverage. It uses this information to identify the areas of overlap, where track reports are likely to be seen from multiple sensors.

The radar geometry can be communicated to SPx-FUSe either through a "site geometry" data file, or else through a set of commands that report radar position and coverage.

Each radar has a position in a world coordinate system, typically defines by its latitude and longitude. Track reports generated from each tracker will be relative to the position of the radar. The combination of the radar's position and the relative position of the target allows SPx-FUSE to calculate the world location of each target, and hence be able to compare positions between radars.

Each radar has a coverage defined by a set of sectors of definable start angle, end angle, start range and end range. Full coverage out to a defined range is represented by a start angle of 0 and an end angle of 360, but in some situations it is necessary to define a radar's coverage as being only in a specific sector (the radar may rotate 360 degrees but be only transmitting in a defined sector, for example).

As an example, Figure 5 shows three radars at different positions. Radar A's coverage is from zero, but is blanked out from angles 180 to 300 degrees. Radar B has full coverage from 0 to 360 degrees. Radar C's coverage starts at a mid range and omits a segment from 170 to 190 degrees.

The definition of radar geometry can include any number of segments.

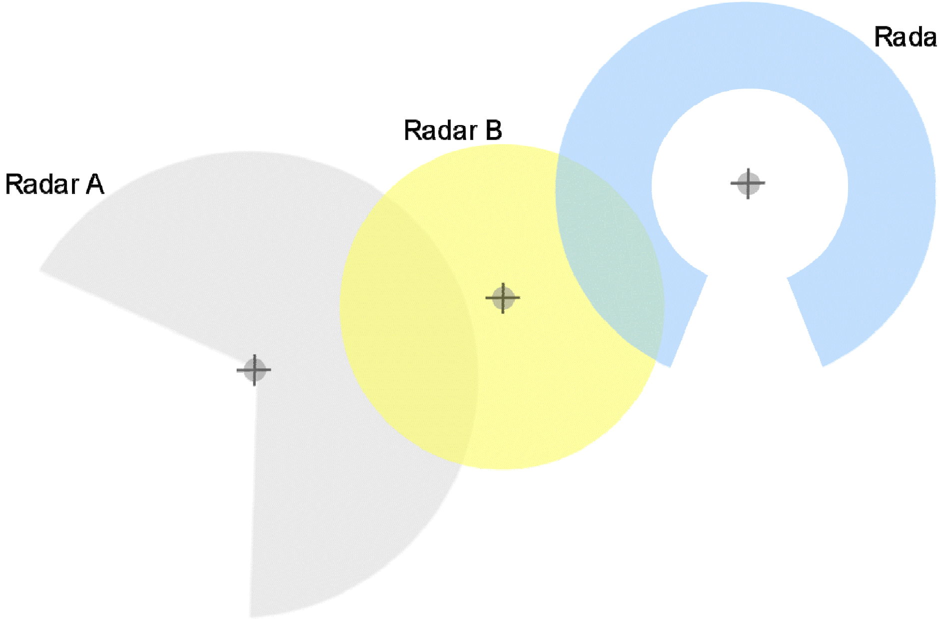

Figure 5 - Sample Radar Geometry and coverage

4.2 Time Stamping

To enable accurate combination of track reports, SPx-FUSE requires each report to be accompanied by an accurate time-stamp. Each remote tracker is assumed to have a GPS-derived time reference, or else be using a network-based time protocol (eg NTP) to maintain synchronization of time references.

4.3 Association

In the situation where a target is seen by more than one radar, SPx-FUSE may receive multiple track reports for the same target. Since the remote radars are not synchronised, the radars do not see the target at the same time, so in general they see it at a different position (assuming the target is moving). This means the first task of SPx-FUSE is to establish whether two reports that describe a target in a similar position are actually observations of the same target. This is the association problem.

Data association is a key stage in a fusion system. It addresses the problem of correctly matching objects together, so that combination of data makes sense. For example, it would make no sense to average the positions from two sensors if the sensors were looking at different objects. Only if the objects are the same (associated) does combination make sense.

SPx-FUSE will *associate* two track reports from different radars if it believes that the reports are observations of the same target. The association problem has to consider the following factors:

- For a moving target, the track reports from different radars will show the target at different positions because the measurements will be taken at different times. A knowledge of how the target is moving must be used to assess whether a position is reasonable given the known motion of the target.

- The measurement process is never perfect, and even when the known motion of the target is used to compensate for the different measurement times, there will never be an exact match. The problem to address is to allow an association where the difference is just due to measurement noise, without compromising the ability of the system to resolve two targets moving closely together.

4.3.1 Association Methods

The SPx-FUSE module contains a number of association methods, including nearest neighbour. Although simple, the nearest neighbour approach is often very effective. A distance measure can be computed based on errors in observable quantities, such as position and size. These can be gated to consider only measurements that are close enough to the expected value. A measurement can be associated with an object if the distance is less than a threshold. Although it is possible to construct programmable thresholds on distance and size, an alternative approach based on the scale-invariant Mahalanobis distance is attractive and employed by SPx-FUSE.

In the situation where multiple targets are being observed by a sensor the tracker has to consider the best overall association of measurements to tracks. The optimal solution to this problem may not derive from considering the best association of each individual track.

4.3.2 Mosaic Combination

The problem of combing track or plot reports from multiple sensors can be eliminating by organising the radar coverage as a mosaic, ensuring that each position is covered by at the most one sensor.

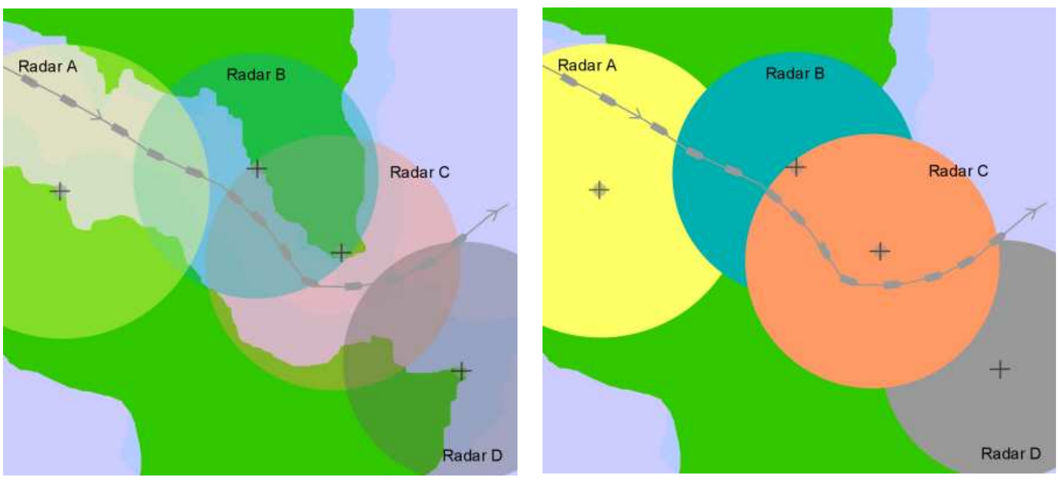

Figure 6 - Mosaic combination of overlapping radars

In Figure 6, the overlapping radar coverage on the left hand picture are combined by assigning a priority to each radar – effectively a measure of the quality of the radar sensor. This defines a preferred sensor at each point. Although the diagram shows the mosaic constructed simple from the priority of the radar, in practice it may be more complex, allowing complex geometric shapes to define the preferred sensor at each location.

4.4 Rules for Combination

When SPx-FUSE has associated tracks it can establish a link between the reports coming from the remote trackers. For example, if track id X from tracker A is associated with track id Y from tracker B, then all future track reports of X and Y can be related to each other – SPx-FUSE has made an association and uses the track ids allocated by the remote tracks as the key to join the reports.

With the association made, each new report for X or Y causes SPx-FUSE to make a decision about how to process the report and what to report itself. The "Rules for Combination" control how SPx-FUSE handles the reports.

SPx-FUSE provides a lot of flexibility for the strategy to combine track reports. The options include:

Preferred sensor. In the situation where two sensors overlap, it might be preferable to use the observation from a preferred sensor that is known to be more reliable. In this situation, the preferred sensor is defined to SPx-FUSE and the other sensors in that area are ignored. Of course as the target moves out of the coverage of the preferred sensor, the non preferred sensors have to take over.

Best Sensor. In the Best Sensor mode, SPx-FUSE makes a judgement about which sensor is providing the best information in the overlap region. The other sensors are ignored, until such time as they are regarded as providing the best information. The decision on best observation is based on historic measurements from the sensors. For example, a sequence of measurements that have a low variability suggest a combination of an accurate measurements process and well-known target dynamics, so that sensor is preferred.

Weighted Sensor. In this mode, the information from multiple sensors is combined with a weighting that is dependent on the historic accuracy of the sensor information. For example, if one sensor has provided very regular measurements, and the other more erratic or missing measurements, then more emphasis is given to the accurate sensor. This is just a more general case of the Best Sensor mode.

Fixed Weighted Sensor. This is just a special case of the weighted sensor mode above, in which the weighting given to each sensor is fixed.

Shift Weight. In this mode, which is another variant of the weighted sensor method, the weight given to two measurements is shifted as the target moves across the intersection. Just as it enters the intersection, the weight is 100% sensor 1. As it leaves the intersection the weight is 100% sensor 2. As it moves between the weight is proportional to the distance, so that at the midpoint the weight is 50% sensor 1, 50% sensor 2.

Range Preferred. In this mode, SPx-FUSE uses the principal that the range measurement of a track is often more accurate than the azimuth estimate. So an improved estimate if position can be made by using range measurements only.

User-defined. There is the option to add your own code to define the combination mode. This code could implement any combination strategy, possibly including some of the built-in modes listed above.

SPx-FUSE defines a set of criteria for measuring the quality of a track report and fusion results. On the assumption that the targets are moving in straight-line with occasional changes of heading, SPx-FUSE calculates the variance in the reported positions as an indication of the quality of the tracking.

The ability for SPx-FUSE to internally monitor the success of its own fusion means it can run a number of fusion strategies in parallel and make a decision on the most effective.

4.5 Format of Incoming Track Reports

The incoming track report from the remote tracker is assumed to contain some or all of the following information:

Track identifier – This is normally a small integer allocated by the remote tracker when the track is first initiated. All track reports relating to the same target should include the same track identifier. When the track is lost, the identifier becomes available again for a new track, although in practice the same identifier would not be used until all other possible identifiers have been employed.

Time-Stamp – An accurate time-stamp needs to be provided with each track report. It is assumed that each remote tracker has its own GPS time source, or else is using a synchronised network time protocol to establish a common reference.

Filtered position – The filtered position is the tracker's estimate of the position of the target based on recent measurements. The filtering process employed by the remote tracker should remove measurement noise by a form of averaging. In practice, the remote tracker may be using an alpha-beta or Kalman filter. The filtered position may be provided as either a range, azimuth value (relative to the radar's location) or else as a lot/long pair. In the event that there was no recent measurement from the sensor, the filtered position would be a prediction of the expected position based on the last known position and estimate speed and heading.

Measured position – The measured position is the most recent observation of the target's position without filtering. This observation is subject to measurement error The measured position may be provided as either a range, azimuth value (relative to the radar's location) or else as a lat/long pair. In the event that there was no recent measurement from the sensor, the measured position is reported as empty.

Speed and course – The estimated speed and course are filtered and derived from the same tracking filter that estimates the filtered position. The values represent the best estimate of the target's speed and direction of motion.

Size – The estimated size of the target is provided as a range and azimuth dimension.

Confidence levels – If the remote tracker provides a measure of track confidence, this can be reported to SPx-FUSE in this information field. A feature of the Kalman filter is that provides estimates of the errors in the state information, so for a remote tracker employing this type of filter returning the error covariance can be provided as a measure of confidence.

Of the above, the essential components are the track identifier, time stamp and one of the positions (ideally measured). Where other information is provided it will help the association and combination processes.

SPx-FUSE is designed to be flexible with respect to input formats. A front-end conversion module is required to receive track reports from a network or serial interface. If the required input format is not a standard currently supported, the module can be written by the user or by CambridgePixel.

4.6 *Format of Outgoing Track Reports*

Like incoming reports, SPx-FUSE allows the output of track reports in a number of different formats. It is easy to write an output module that reformats the data into a new report style.

4.7 *Handling Track Identifiers*

A track identifier is a handle that a tracker uses to identify a track. In a single-channel tracker, a new identifier is allocated when a new track is initiated, and subsequent updates of the track generate new track reports with the same identifier. Often the track identifier is a small integer which uniquely identifies a track in a tracker, but is not unique across multiple trackers. For example, in a multi-tracker configuration, it would be expected that remote trackers A and B would both be processing tracks with the same identifier. This is not a problem since the combination of tracker id and track id uniquely identifies the track.

After combination of tracks in SPx-FUSE it is desirable that there is a single track identifier presented on the output. This is of biggest concern when a track needs to transfer from tracker A to tracker B.

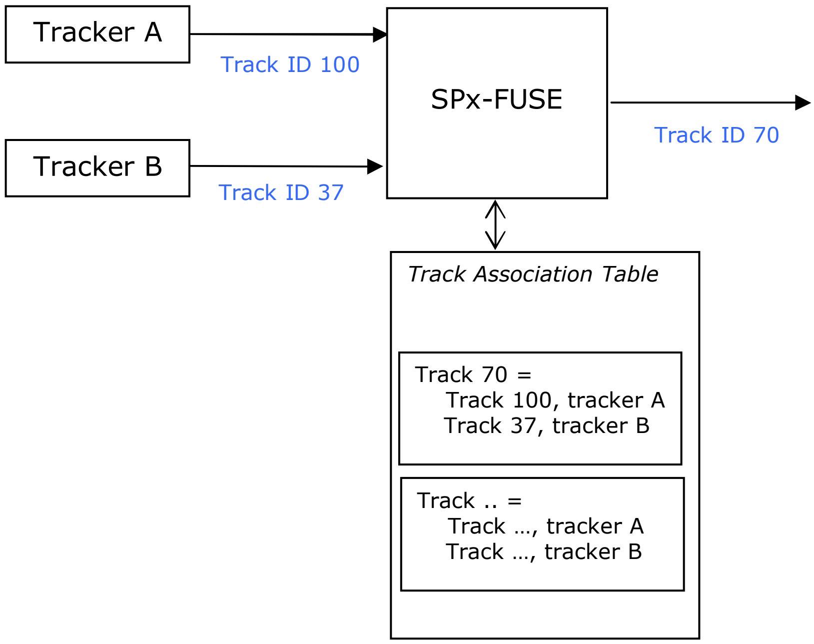

Figure 7: Handling track identifiers

In Figure 7 Tracker A generates a report for track 100, which is believed (by SPx-FUSE) to be the same target as track 37 in Tracker B. This "belief of association" is maintained by SPx-FUSE as an entry in a track association table. SPx-FUSE has allocated an identifier 70 to represent the track that is known to tracker A as 100 and to tracker B as 37. The track reports that are output from SPx-FUSE will only feature the id of 70 and the individual tracker ids of 37 and 100 are regarded as "internal" only. It is worth noting that the id of 70 is allocated by SPx-FUSE when the *first* track report is received; let's say that's id 100 from tracker A. At that point, it is likely that there is no output from tracker B (the target isn't seen by it), and for some time SPx-FUSE simply receives one set of track reports and converts the incoming id of 100 into the output id of 70.

Of course the ids of 37, 70 and 100 in the above example are arbitrary numbers and serve only to demonstrate the point.

Let's summarise this with the following sequence of events:

1. Tracker A sees a target and generates a track report with id 100.
2. SPx-FUSE sees the track report and generates a new entry in its track database with id 70, recording the association with track 100 in tracker A.

3. Tracker A continues to generate reports for track 100 and sends these reports to SPx-FUSE.

4. SPx-FUSE continues to receive reports from tracker A and generate output reports with ID 70.

5. At some point, tracker B sees the same target and outputs a track report with id 37.

6. SPx-FUSE receives the track report from tracker B and determines an association with the existing track 100 from tracker A. The track database is updated to record the relationship with SPx-FUSE's track 70, track 100 from tracker A and track 37 from tracker B.

7. As track reports come in for track 100 in tracker A and track 37 from tracker B, SPx-FUSE uses the defined combination rule to construct the parameters of the output track 70.

8. At some point the target goes out of range of tracker A and tracker A stops sending reports. SPx-FUSE detects this and updates its association table to remove the link.

9. At some point the target goes out of the range of tracker B and the competion of reports from that tracker causes SPx-FUSE to delete the tracker from its database.

4.8 *Timing of out-going tracker reports*

In the situation where a target is being covered by two or more sensors, SPx-FUSE will receive track reports at a rate equal to the sum of each individual tracker. In the simple situation of trackers A and B generating reports on a single target at a rate of 1 per second (60rpm scan rate of the radar), SPx-FUSE receives track reports at the rate of 2 per second, one from each tracker. The question arises as to how SPx-FUSE should handle this.

It should be noted that there is, in general, no fixed relationship between the timing of the reports from trackers A and B for the same target. Although they may nominally have the same period of update, the time difference between the updates in not fixed and will change.

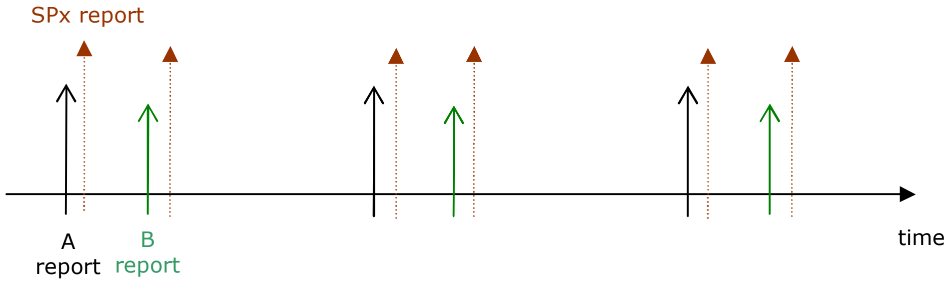

Figure 8: Timing of track report generation

The default behaviour of SPx-FUSE is that it tries to generate a track report for each relevant input report that it receives. In the case of Figure 8, the interval between reports from tracker A is approximately constant, as is the interval between report from tracker B. But the time difference between the A and B reports is not specified and will change. The SPx-FUSE reports happen at each update. SPx-FUSE will not generate a track report if the report would be the same as the previous one. For example, if the current track fusion rules are set to use tracker A in favour of tracker B for an overlapping situation (Preferred

Sensor rule in Section 0), then SPx-FUSE only creates track reports on updates from tracker A.

4.9 Handling Sensor Failure

One of the justifications for using multiple overlapping sensors might be to provide some redundancy in the event of a sensor failure. SPx-FUSE handles the situation of a sensor that stops providing data by switching to an alternative sensor where possible.

4.10 Using SPx-LOGIC

The optional SPx software module SPx-LOGIC provides a set of companion capabilities for SPx-FUSE. The GeoLOGIC module provides geometric-based logic processing of track reports. Rules can be configured that provides alerts if a target moves into a designated safe area, or of targets get too close together, or are on a collision course. SPx-LOGIC allows the construction of complex criterion about the relative position of targets compared to one another and to other features in the world. Alerts are automatically generated if rules are broken.

When using SPx-LOGIC with SPx-FUSE, the consolidated track reports from SPx-FUSE are input to the OGIC module for analysis. Full details on the capabilities of the SPx-LOGIC module are contained in separate documentation.

Sample rules for SPx-LOGIC include:

- Generate an alarm if two targets get within a specified distance of each other.
- Generate an alarm if a target goes into an exclusion area.
- Generate an alarm if two targets are on a collision path in a specified time.
- Generate an alarm if two targets occupy the same restricted area (eg airport runway).

5 SPx-FUSE Visualisation

A separate program, SPx-FUSE-Visualise, provides a graphical user interface that provides a window into SPx-FUSE for configuration, status and maintenance purposes. SPx-FUSE- Visualise may run on the same processor as SPx-FUSE, or may run elsewhere, communicating to SPx-FUSE over standard Ethernet protocols. The Visualisor can connect to an already running SPx-FUSE process with no need to restart anything.

The Visualisor provides for display of the following:

- Outline map display of the overall area, showing radar positions and coverage
- Display of track reports from each remote tracker
- Display of track reports output from SPx-FUSE
- Detailed view of a selected track, showing details of track association and combining
- Status information
- Ability to interact with SPx-FUSE to change operating parameters