Multi-sensor Fault Detection, Exclusion and Readmission

Birendra Kujur, Samer Khanafseh, Boris Pervan, *Illinois Institute of Technology*
Andrey Soloviev, *QuNav*

BIOGRAPHY

Birendra Kujur is currently a PhD candidate in Mechanical and Aerospace Engineering at Illinois Institute of Technology. He received his Bachelor of Science in Mechanical Engineering from Purdue University in 2014. His research interests include multi-sensor navigation systems and navigation integrity monitoring. Currently, he focuses on detecting GNSS spoofing attacks and developing anti-spoofing solution.

Dr. Samer Khanafseh is currently a research associate professor at Illinois Institute of Technology (IIT), Chicago, and the principal of TruNav LLC. He received his MSc and PhD degrees in Aerospace Engineering from IIT in 2003 and 2008, respectively. Dr. Khanafseh has been involved in several aviation applications such as Autonomous Airborne Refueling (AAR) of unmanned air vehicles, autonomous shipboard landing for NUCAS and JPALS programs and Ground Based Augmentation System (GBAS). His research interests are focused on high accuracy and high integrity navigation algorithms, cycle ambiguity resolution, high integrity applications, fault monitoring and robust estimation techniques. He was the recipient of the 2011 Institute of Navigation Early Achievement Award for his outstanding contributions to the integrity of carrier phase navigation systems.

Dr. Boris Pervan is a Professor of Mechanical and Aerospace Engineering at IIT, where he conducts research on advanced navigation systems. Prior to joining the faculty at IIT, he was a spacecraft mission analyst at Hughes Aircraft Company (now Boeing) and a postdoctoral research associate at Stanford University. Prof. Pervan received his B.S. from the University of Notre Dame, M.S. from the California Institute of Technology, and Ph.D. from Stanford University. He is an Associate Fellow of the AIAA, a Fellow of the Institute of Navigation (ION), and Editor-in-Chief of the ION journal NAVIGATION. He was the recipient of the IIT Sigma Xi Excellence in University Research Award (2011, 2002), Ralph Barnett Mechanical and Aerospace Dept. Outstanding Teaching Award (2009, 2002), Mechanical and Aerospace Dept. Excellence in Research Award (2007), University Excellence in Teaching Award (2005), IEEE Aerospace and Electronic Systems Society M. Barry Carlton Award (1999), RTCA William E. Jackson Award (1996), Guggenheim Fellowship (Caltech 1987), and Albert J. Zahm Prize in Aeronautics (Notre Dame 1986).

Dr. Andrey Soloviev is a Principal at Qunav. Previously he served as a Research Faculty at the University of Florida and as a Senior Research Engineer at the Ohio University Avionics Engineering Center. He holds B.S. and M.S. degrees in applied mathematics and physics from Moscow Institute of Physics and Technology and a Ph.D. in electrical engineering from Ohio University. His research interests focus on all aspects of GNSS signal processing and estimation, as a well as multi-sensor fusion for navigation applications. He is a recipient of the ION Early Achievement Award and the RTCA William E. Jackson Award.

ABSTRACT

In this work, we present a fault detection, identification, exclusion, and readmission method in a multi-sensor navigation architecture. The example multi-sensor navigation architecture used in this work comprises of different sensors coupled with an inertial measurement unit (IMU) in individual Kalman filters (KF), where each filter separately provides its navigation solution. These decentralized individual KF solutions are later combined to provide the resultant navigation solution using a weighted fusion. In contrast to a centralized KF where faulty sensor measurements corrupt the navigation solution due to KF memory, the decentralized structure allows fault(s) to be removed from the resultant navigation solution instantaneously after detection and exclusion of the faulty filter(s). Solution separation among subsets of filters is used to detect faults and then identify the faulty sensors or filters, after which they are excluded from the resultant navigation solution. We also provide protection levels against faulty filters, which are identified and excluded. In order to readmit the declared faulty filters, we utilize solution separation techniques again to design a readmission test. This readmission test also allows us to reset a faulty filter in case the fault caused the filter navigation solution to be corrupted. We provide integrity risk and protection level formulas that account for missed detection, and wrong readmission. We validate the developed algorithm for detection, identification, exclusion, and readmission of faulty sensor using real sensor data set and simulated fault scenarios.

I. INTRODUCTION

Navigation using multiple sensors has been proposed and utilized previously for its advantages of providing better accuracy and continuity through redundancy. Multi-sensor navigation architecture although beneficial also introduces additional systems that could be prone to fault, and thus integrity monitoring of sensors is necessary if the desired accuracy needs to be achieved at all times. Using multi-sensor integration in safety critical applications such as driver-less cars, autonomous drones etc., calls a need for robust fault detection, exclusion and readmission method. Commonly sensor fusion of a multi-sensor navigation architecture comprises of using a single centralized KF to combine measurements from different sensors. A big drawback of such kind of fusion method is that once a fault occurs in one of the sensors, even after detection and exclusion of such faulty sensor from the measurement, the effect of fault prior to exclusion still remains in the KF solution due to KF memory. To alleviate such problems, we propose to utilize a decentralized filter architecture where each sensor has its own measurement filter such as KF, which provide decoupled individual navigation solution. Each of these solutions is weighted according to the covariance of the sensors and combined together to obtain snapshot resultant navigation solution. This resultant solution is memoryless, and hence if a faulty filter is excluded after detection, the effect of fault is instantly removed from the resultant navigation solution in the next epoch. Integrity methods using variety of tests have been proposed for fault detection and exclusion in a multi-sensor architecture (Kerr (1987); Nikiforov (2002)). In this work, we propose a complete fault detection, identification, exclusion and readmission method based on solution separation test. Solution separation has been widely used in aviation applications for detecting satellite faults, and is the baseline monitoring architecture for ARAIM program (EU-U.S. Cooperation on Satellite Navigation Working Group C - ARAIM Technical Subgroup (2016)). Solution separation is attractive because it measures the effect of the fault on the position domain (or any specific state(s)) directly. As a result, it does not require defining a specific temporal or spatial fault profile. This attractive feature makes the evaluation of the system integrity risk valid for any type of fault without the need to run many simulations to cover all types of faults the system may have. Methods of fault detection and identification have been previously developed (Zhai et al. (2018)), which we adopt and extend for the case of sensor fusion. However, most of the referenced prior work, like in ARAIM, uses snapshot navigation solutions (least squares, for example). In KF implementations, it is not enough to not use the faulty sensor or measurement at the time of fault detection. Since KF has memory of past measurements, by the time the fault caused the test statistic to exceed the threshold, it may have already corrupted the filter. Thus, removing the fault at that epoch may not be sufficient for the solution to be deemed fault free and so we need to check for faults after each measurement update.

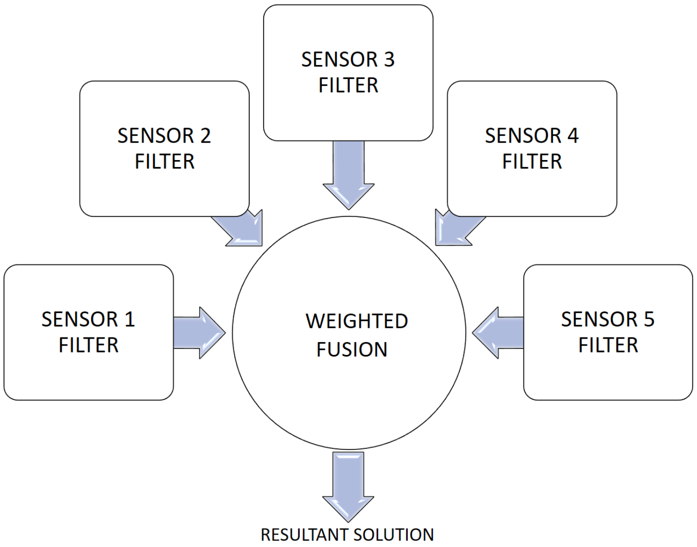

Figure 1: *Decentralized structure for sensor fusion.*

We utilize the federated filter approach as the architecture of our multi-sensor navigation (Carlson and Berarducci (1994); Hashemipour et al. (1998); Mutambara (1998)), where each individual parallel KF (also known as federated filters) processes a measurement from each sensor separately. These filters are later combined to form the full set resultant navigation solution. Figure 1 illustrates this decentralized structure for sensor fusion where individual KF solutions from each sensor filter is combined based on covariance weighting to produce a snapshot resultant solution. For detection we compare the combined full set resultant navigation solution (namely position and velocity states) to subset solutions constructed by combining subsets of the individual filters. A fault is detected when the difference between the resultant navigation solution and subset outputs exceeds a threshold. The algorithm then proceeds by applying the detection step to the subsets (by forming solutions using sub-subsets) in order to find a subset that includes the maximum number of individual filters while being consistent. This identification test enables us to confirm that the identified filters are the faulty ones, and then exclude them from the resultant navigation output.

Once a faulty sensor is identified, it is not used in the resultant navigation solution, but the filter still continues running in the background in a "quarantine" mode. This allows the system to readmit the filter, in a false alarm event, without losing the information accumulated in the filter's history. For readmission, this identified filter is still being checked in a readmission solution separation test. A quarantined sensor is not used unless the solution separation test deems it fault free. In case the fault causes the filter navigation solution to be corrupted for a period more than a specific threshold, the system invokes a filter reset. This reset is implemented to identify whether the fault in the filter solution is caused by sensor fault that still exists, or due to a latent fault that does not exist anymore, but due to filter corruption when the fault was present. In this work, we also provide a formula for evaluating the integrity risk and protection levels due to wrong readmission in this logic. We also present the complications and constraints of this system in detecting multiple simultaneous faulty sensors or filters. Section II details the fault detection, identification and exclusion method, and in section III we introduce the filter reset and readmission algorithm. In section IV we explain the integrity risk and method of protection level computation pertaining to detection, identification, and wrong readmission. The results using real measurements from sensors with simulated faults are shown in section V. Finally, the appendix at the end provides all the sensor specifications.

II. FAULT DETECTION, IDENTIFICATION AND EXCLUSION

Due to the nature of solution separation tests being utilized for fault detection, identification and readmission, we need enough redundant filters to detect multiple faulty filters and eventually readmit them once faults no longer exist. This is known as multiple fault hypothesis testing. The decision to test the hypothesis of multiple filters being faulty at any instant of time is given by the prior fault probabilities. Suppose sensor A, B, and C all have independent prior fault probability of 10^{-6} , and thus fault probabilities of single, dual and three filters to be at fault in any time instance is 10^{-6},10^{-12} and 10^{-18}. A user might opt to not check for hypothesis that three filters can be at fault at any given instance of time since the probability for that hypothesis to be true is negligible. For testing multiple fault hypothesis using solution separation in conventional systems, the more sensors are available to the user, the higher the computation load will be due to running multiple parallel filters. For example, consider a case where the system uses 5 sensors. If all subsets are to be considered for fault detection, including multiple sensor faults, the number of subset solutions that need to be formed are $C(5,0)$ (all sensors) + $C(5,1)$ (for 1 sensor fault subsets) + $C(5,2)$ (for 2 sensors fault subsets) + $C(5,3)$ (for 3 sensors fault subsets) + $C(5,4)$ (for 4 sensors fault subsets) = 31, where, C is the combination operator.

In principle, all these computations would need to be executed at each measurement epoch. However, unlike central filter implementations where 31 filters need to be running simultaneously, using federated filter approach, only 5 KF are used (one for each sensor), and the 31 required solution combinations is formed using individual filter process. Thus, the computation load is expected to be orders of magnitude less than running 31 parallel filters. If any of these 31 solution separation tests triggers an alarm, it is an indication that one (or multiple) filters are faulty. However, the fault source cannot be known reliably by only considering what tests triggered the alarm and we need further tests for identification. For identification if n-fault hypothesis needs to be tested, then in general we need $(2n+1)$ filters to reliably identify faults. In other words, if we have total of m filters then we can identify maximum of $((m-1)/2)$ fault hypothesis. Thus, for 5 filters we can identify a maximum 2 simultaneous fault hypothesis. For our case we consider total number of filters as 5 and a maximum of dual fault hypothesis. The detection process first tests for maximum fault hypothesis (here dual) and then identifies the faulty filter using single fault hypothesis. First, the detection test is composed by comparing the weighted resultant output using all 5 sensors to a subset that is composed of 3 sensors (since we test for hypothesis that two sensors are at fault). For example, we take full filter set [1 2 3 4 5] and compare against [3 4 5] if we assumed [1 2] to be faulty. This leads to maximum solution separation tests of $C(5,2)$ (for 2 sensors fault subsets) = 10. If detection is triggered we know that there is a fault either in one of the sensors or both sensors. Also, due to the combinations of sensors, more than one subset could trigger.

To identify the faulty filter, we start with the subset whose normalized test statistic is the maximum. This subset with 3 filters is compared to all combinations of sub-subsets with 1 sensor i.e a self consistency solution separation check is performed for the 3 filter subset since a case could happen where two filters are faulty and one of them is in the subset. Thus, in the above example if fault was detected for combination set of [3 4 5], we would perform consistency solution separation check by comparing [3 4 5] against [3 4], [4 5] and [3 5]. If this check fails, then we go to the next combination obtained during detection. Once consistency check is valid, we perform solution separation of each filter of the fault hypothesis one at a time. That is, taking the above example and comparing [1 3 4 5] against [3 4 5] and then comparing [2 3 4 5] against [3 4 5]. Since [3 4 5] has been established as fault free, if both tests fail then we have two filters (here both [1 2]) that are faulty and if only one of them fails then we have just one faulty filter (here either 1 or 2 based on which test failed). If none of them fails, then we take the next combination say 3 and 4 to perform the solution separation test. Once identified we proceed to exclude this faulty filter(s) and then wait for next measurements to apply the readmission algorithm.

III. FILTER READMISSION AND RESET

Once the faulty filter is excluded from the weighted resultant solution, it runs in the background with a readmission process testing if in case it becomes fault free with every new incoming measurements. We again rely upon the solution separation method to test whether a prior faulty filter is fault free. Depending on prior fault probabilities, the readmission solution separation test subsets are chosen. For example if only one faulty filter was excluded but the fault probability of dual fault hypothesis is significant then different cases can arise as explained below.

Case 1: Some faulty filters are yet to be excluded during readmission of faulty filters

In this case, the hypothesis is that there were originally two faulty filters out of which the first faulty filter was detected and excluded while the second filter is faulty, but not detected yet. During this intermittent time when the second filter is yet to be detected, the first filter passes the readmission test and is readmitted. Here there could be a case in which the yet undetected second filter conspired against the system by prematurely readmitting the first faulty and excluded filter (hence wrong readmission).

Case 2: All faulty filters are excluded, and readmission occurs at different time epochs

In this case, the hypothesis is that the faulty filters have been excluded but readmission of the faulty filters (that are still faulty) happen at different times. In this case, the integrity of the system is jeopardized if a faulty filter is wrongly (or prematurely) readmitted to the system, influencing later readmissions, and causing a further premature readmission of the other filter. Such a scenario may happen when the wrongly readmitted filter causes the navigation solution of the assumed fault free subset to be closer to the excluded faulty filter, and thus, causing the test statistic to not exceed the threshold. Since in the case of wrongly readmitting a filter, causing the test statistic to be smaller, we will assume that wrongly readmitting a filter, while others are still excluded, always increases the probability of wrong readmission of other excluded filters.

Case 3: Readmission for all faulty filters occurs at same time epoch

For this case, we implemented readmission such that it only allows a single filter to be readmitted at a time and therefore, becomes similar to Case 2. The first filter to be readmitted would be the one which has the smaller normalized test statistic.

Given these different cases of wrong readmission, we again rely upon prior fault probabilities of sensors. Thus, regardless of the number of filters excluded after fault we always perform readmission solution separation test based on the maximum fault hypothesis. That is, for example if we consider dual fault hypothesis as our maximum fault hypothesis (which we used for detection as well), we will form solution separation subsets of three given the total number of filters is five. Thus, the readmission solution separation test becomes exactly as the one used for detection.

One of the reasons why a filter cannot be readmitted right after the fault disappearance is the memory effect of the fault on the filter. Meaning, although the fault disappeared, its effect on the filter and state vector may linger. One way to erase the fault from the filter's memory is to reset it. Without filter reset, there could be cases where the filter is slow in being readmitted. For this reason if a filter remains excluded for a predefined time the filter is being reset. For example, in the case of a dead reckoning filter, like 2D laser, once subjected to a fault for long periods, it will not correct itself after the fault ends. Therefore, a filter reset is useful in eliminating the effect of the latent fault.

IV. INTEGRITY RISK AND PROTECTION LEVELS

Integrity risk is defined as the probability of hazardous misleading information $P(HMI)$, which is probability of estimate error ε exceeding alert limit L, and can be written as,

$$P(HMI) = P(|\varepsilon| > L) \tag{1}$$

When the monitor is testing for a fault in a filter, two mutually exclusive and exhaustive events can occur, which are: f when fault is present, and nf when no fault is present. Using the law of total probability, we can write equation (1) as,

$$P(HMI) = P(|\varepsilon| > L|f)P(f) + P(|\varepsilon| > L|nf)P(nf) \tag{2}$$

Given that the navigation system has the solution separation (SS) monitor that computes a test statistic q_{ss} and threshold T_{ss} to detect faults, we can write equation (2) as,

$$P(HMI_{SS}) = P(|\varepsilon| > L, q_{ss} < T_{ss}|f)P(f) + P(|\varepsilon| > L, q_{ss} < T_{ss}|nf)P(nf) \tag{3}$$

The second term in equation (3) is the fault free integrity risk with fault free hypothesis H_0. The first term can be expanded for multiple fault hypothesis. Also, if an integrity budget of $P(HMI_{SS})$ is given, then we can substitute the alert limit L with protection level PL. Thus, equation (3) can be written as,

$$P(HMI_{SS}) = P(|\varepsilon| > PL|H_0)P(H_0) + \sum_{i=1}^{m} P(|\varepsilon| > PL|H_i)P(H_i) + \sum_{j=1}^{m}\sum_{k=1}^{m} P(|\varepsilon| > PL|H_j \cap H_k)P(H_j \cap H_k) + \ldots; j \neq k, \qquad (4)$$

where, m is the total number of filters, $P(H_i)$ is the probability of fault in sensor i. The first term in equation (4) is the fault free integrity risk and the second and third term represent integrity risk due to single and dual fault hypotheses, respectively, and the equation can be expanded further for multiple fault hypothesis. Recall, in this work, we restricted ourselves to dual fault hypothesis and so we are only concerned with the first three terms of equation (4). In this work we implement the PL computation that has been utilized by the Advanced Receiver Autonomous Integrity Monitoring (ARAIM) algorithm (Cassel (2017)). The idea is to determine the optimal PL based on the total integrity budget $P(HMI_{SS})$ instead of allocating the budget to each hypotheses. For PL computation we use $P(H_i)$ as 10^{-6}. Since the readmission solution separation test is the same as detection, the protection level computation for wrong readmission is the same as that of missed detection.

V. RESULTS AND DISCUSSION

We tested our algorithm with real data from six sensors namely IMU, GPS, odometer, barometer, 2D laser and 3D laser. A ground vehicle equipped with all these sensors was driven along a closed loop trajectory (Figure 4) to collect data. Table 1 shows the measurement frequency for all the utilized sensors and Appendix A lists the sensor specifications. The IMU is tightly coupled with GPS, odometer, 2D laser, barometer and 3D laser to form individual federated filters. These filters each provide their individual navigation solution which are combined to get the resultant navigation solution.

Table 1: Measurement rate of sensors

Sensor	IMU	GPS	Odometer	2D laser	Barometer	3D laser
Frequency (Hz)	100	1	1	5	25	2

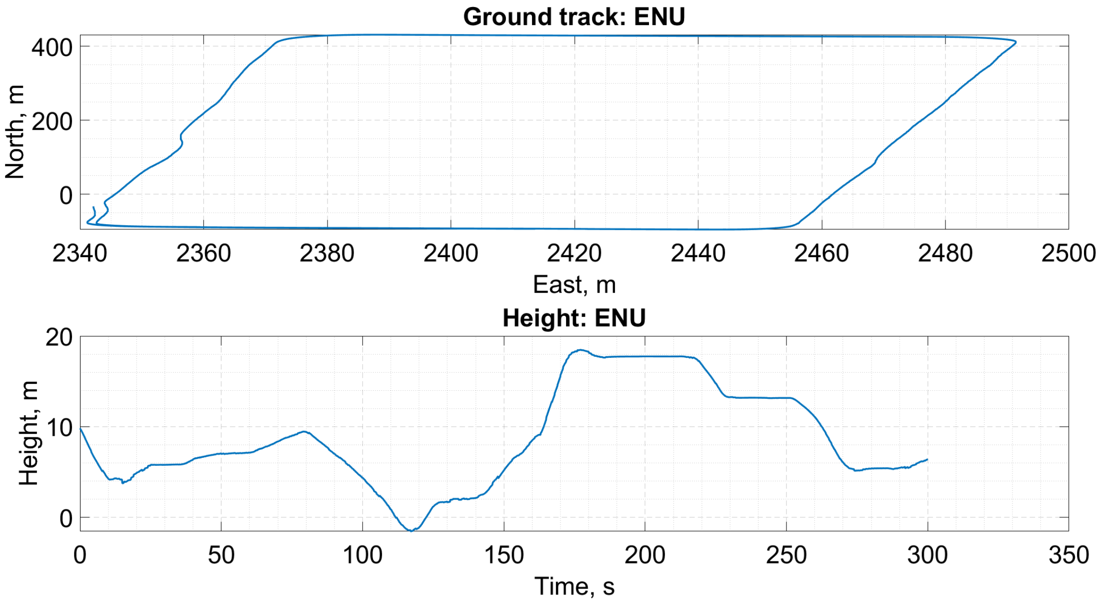

Figure 2: *True trajectory of the ground vehicle used to collect sensor data.*

First in order to illustrate the importance of integrity monitoring of sensors in this decentralized multi-sensor fusion, we compare the horizontal position errors of the resultant solution with and without the solution separation monitor. Figure 3 illustrates the horizontal position error for the resultant solution when GPS ramp fault exists. When solution separation monitor is enabled, it detects and excludes the GPS filter from the resultant solution such that effect of GPS fault is removed. On the other hand when no integrity monitoring is employed, the horizontal position errors of the resultant solution breach the assumed fault free protection level.

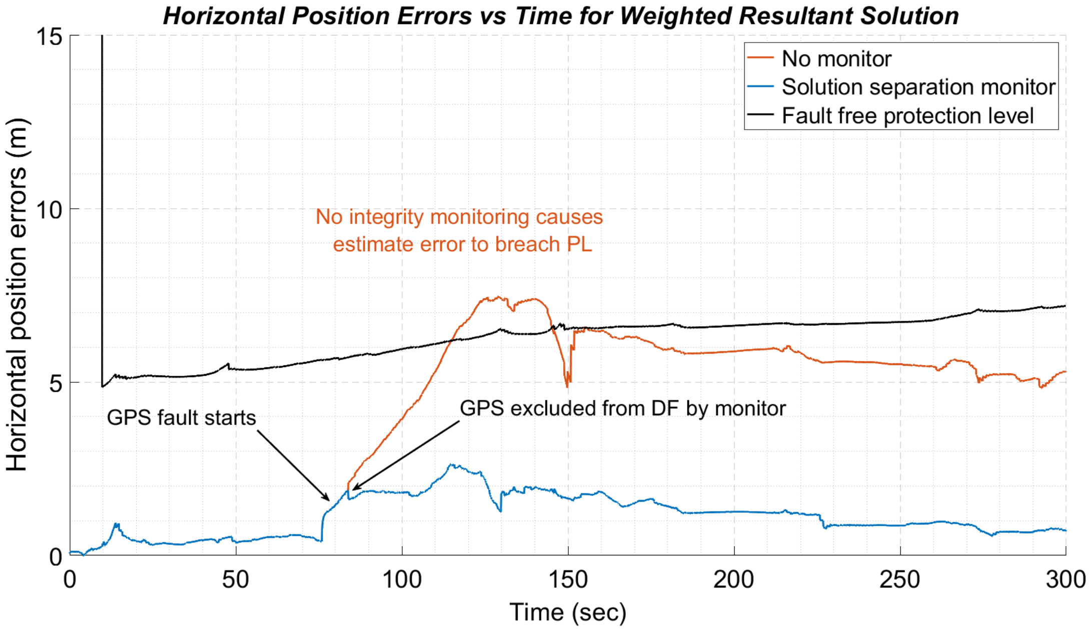

Figure 3: *Horizontal Position Errors versus Time for Weighted Resultant Solution (GPS Fault).*

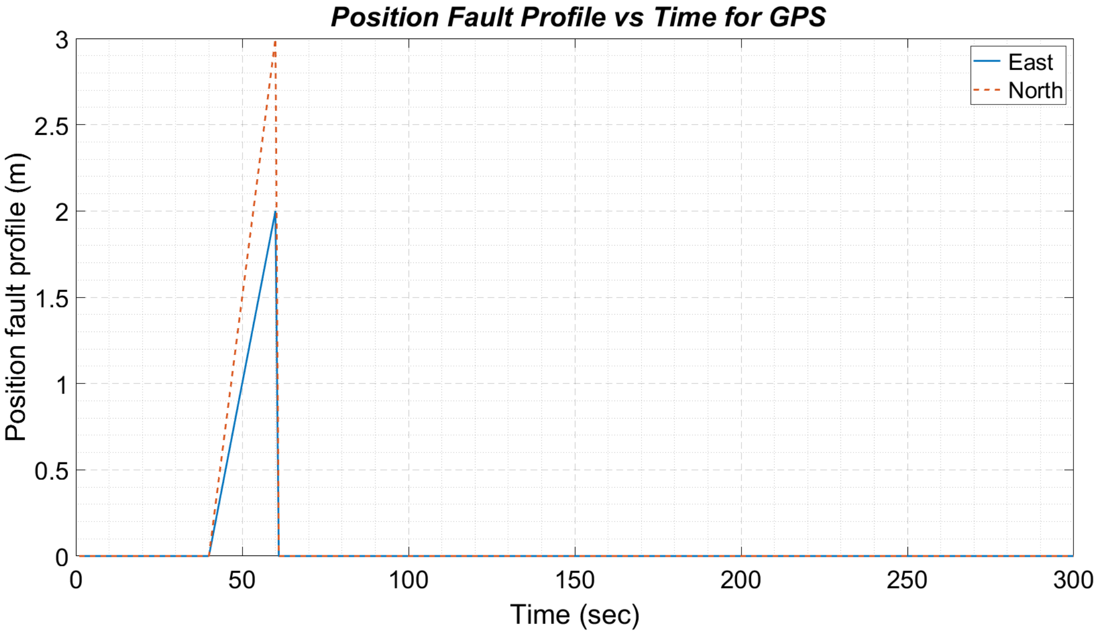

Figure 4: *Position Fault Profile in East and North Direction for GPS Sensor.*

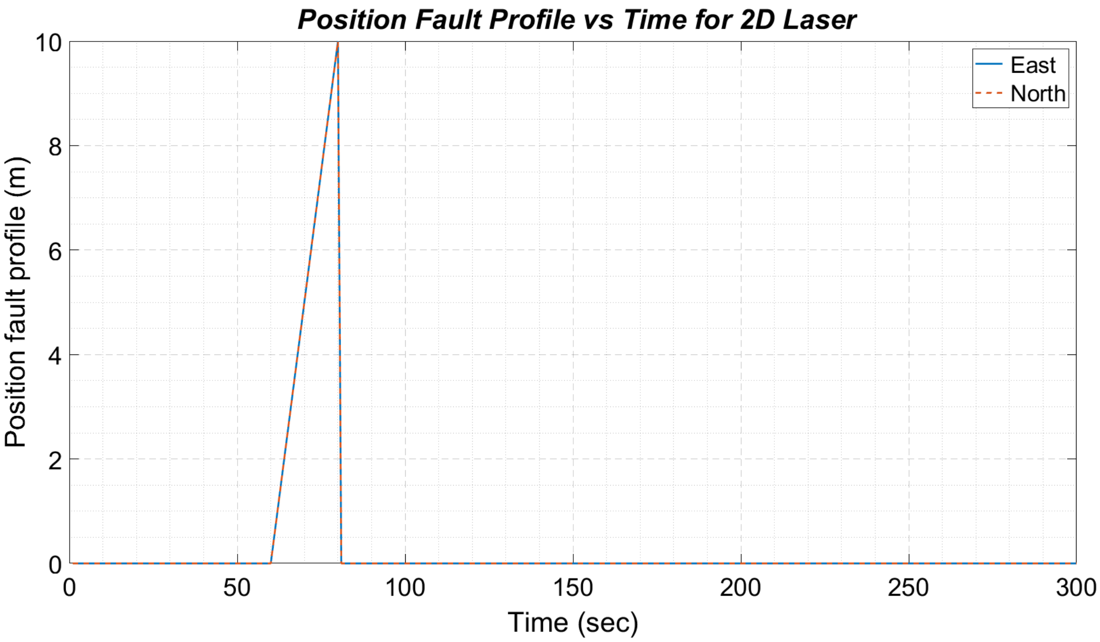

Figure 5: *Position Fault Profile in East and North Direction for 2D Laser Sensor.*

To validate the performance of our integrity monitor, we show results for two test scenarios, first when a fault is injected to a single sensor (2D laser) and second when the fault is introduced to two sensors (GPS and 2D laser). We injected position ramp faults to the GPS and 2D laser sensors, where Figure 4 and Figure 5 illustrate the horizontal position fault profiles for these two sensors, respectively.

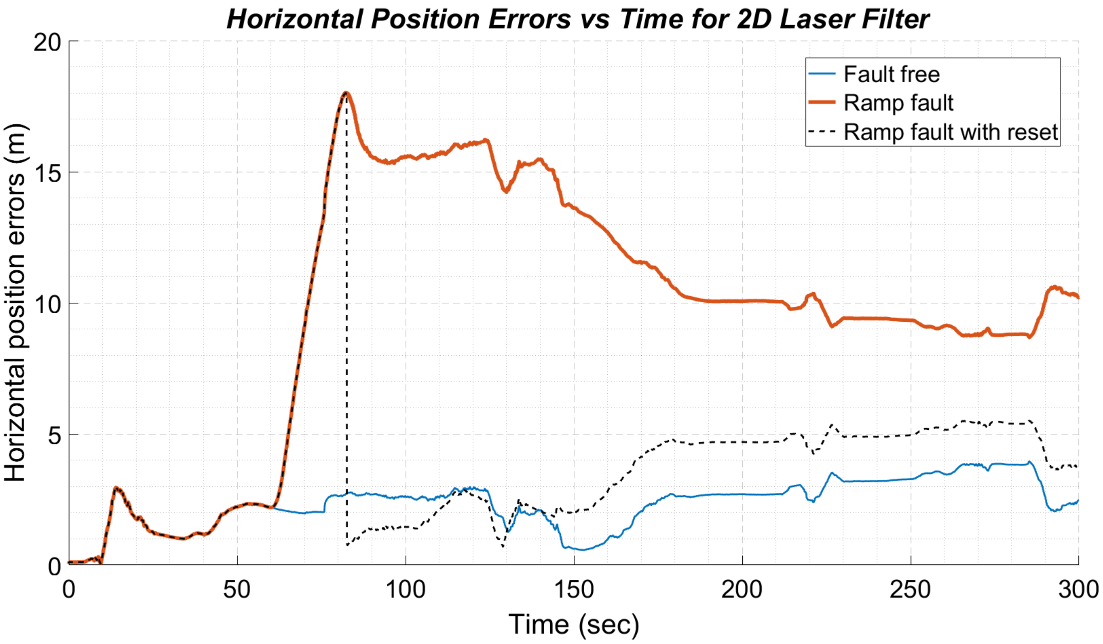

Figure 6: *Horizontal Position Errors versus Time for 2D Laser Filter (2D Laser Fault).*

For the first scenario we introduced horizontal position fault to the 2D laser sensor as illustrated in Figure 5. Since 2D laser is a dead reckoning type sensor, the fault introduces a bias in the filter position error which cannot be corrected by itself, thus making the case for filter reset. Figure 6 illustrates the horizontal position error for the 2D laser filter with fault and reset after the fault, in comparison to the fault free scenario. This fault mimics a case where the 2D laser system miscalculates the amount the user moved due to landmark mis-association between two consecutive snapshots, for example.

Figure 7 illustrates the horizontal position error for the resultant solution for this 2D laser fault scenario in comparison to fault free scenario. As soon as the faulty filter is identified, it is removed from the resultant solution and readmitted after 150 s when it passes the solution separation readmission test. In contrast when the 2D laser filter is reset, the fault bias is corrected and it is immediately readmitted to the resultant solution. It is clear from the figure that not only filter reset corrects any bias fault in the individual filter, it allows for tighter protection level and early continuity to the navigation system.

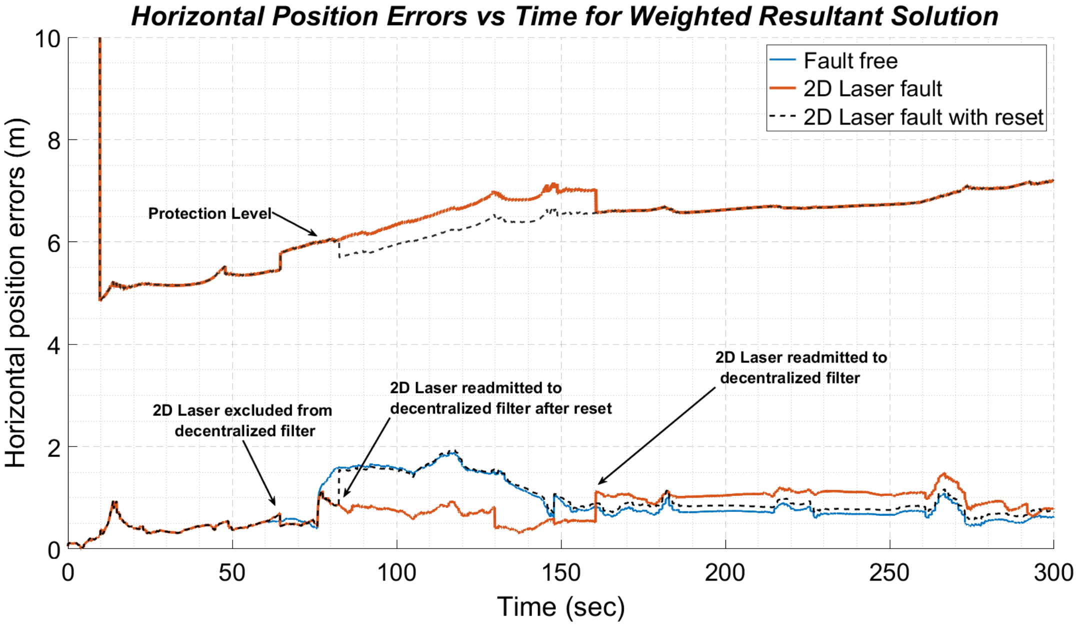

Figure 7: *Horizontal Position Errors versus Time for Weighted Resultant Solution (2D Laser Fault).*

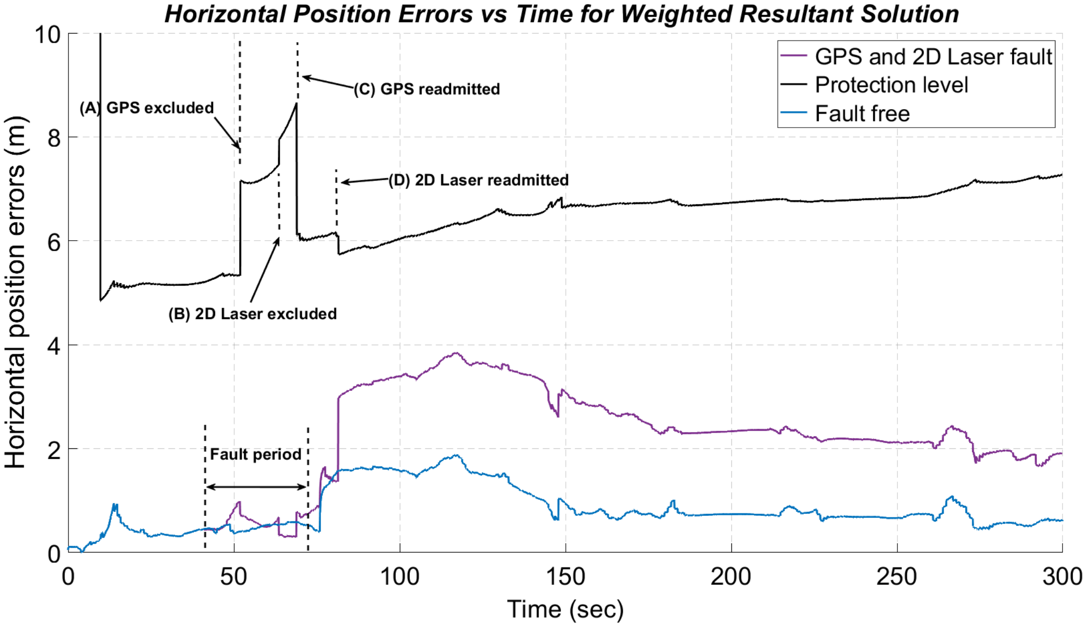

Figure 8: *Horizontal Position Errors versus Time for Weighted Resultant Solution (GPS and 2D Laser Fault).*

For the second scenario we introduced fault to both GPS and 2D laser with reset of 2D laser filter after fault detection. Figure 8 illustrates the horizontal position errors for the resultant solution for this scenario of two faults in comparison to fault free scenario. First fault is injected into GPS and then into 2D laser. The GPS fault is detected and it is excluded from the resultant solution with protection level increasing accordingly. Then 2D laser fault is also detected with exclusion of 2D laser from the resultant solution. We observe a respective increase in protection level due to this exclusion. Once GPS fault ends and it is deemed fault free by the solutions separation monitor, it is readmitted to the resultant solution. Similarly for 2D laser, after the reset it is readmitted to the resultant solution. The relative weighting of the sensors for estimating the position in the resultant solution contributes to how the faults affect the resultant solution. Although GPS is readmitted to the resultant solution, the GPS filter memory still has residual fault in it (but less than the detection threshold). The GPS filter is weighted more than the 2D laser filter and hence the effect of GPS fault on horizontal position errors of resultant solution is seen as a bias in position errors which is slowly converging to fault free position errors. The weighting of GPS being more that the 2D laser also affects protection level. This is obvious in the protection levels as illustrated in the figure when the effect of GPS filter exclusion on protection level is worse as compared to 2D laser.

VI. CONCLUSION

We proposed a decentralized multi-sensor weighted fusion architecture, where as an example individual sensors tightly-coupled with an IMU inside a KF architecture provide individual navigation solution. The navigation solution from all these individual filters are weighted and combined to form the resultant solution. The individual navigation solutions and their combinations are compared to the resultant solution in a solution separation test to detect and identify faulty filters, which are then excluded from the resultant solution. The readmission of declared faulty sensors also utilizes solution separation test with a reset option if the filter is deemed to have been corrupted by the fault. We also evaluated the navigation solution integrity risk due to missed detection, missed identification and wrong readmission which is then used to calculate the respective protection levels. Lastly, we demonstrated the proposed algorithm using real sensor data with simulated faults.

VII. ACKNOWLEDGEMENT

The authors gratefully acknowledge the US ARMY for supporting this research under the STTR program. However, the views and opinions expressed in this paper are those of the authors alone, and do not necessarily reflect the opinions of any other organization or person.

REFERENCES

Carlson, N. A. and Berarducci, M. P. (1994). Federated kalman filter simulation results. *Journal of The Institute of Navigation*, 41(3):297–321.

Cassel, R. (2017). Real-time ARAIM using GPS, GLONASS, and GALILEO. *M.S. Thesis*.

EU-U.S. Cooperation on Satellite Navigation Working Group C - ARAIM Technical Subgroup (2016). Milestone 3 Report. Technical report, ARAIM.

Hashemipour, H. R., Roy, S., and Laub, A. J. (1998). Decentralized structures for parallel kalman filtering. *IEEE Transaction on Automatic Control*, 33(1):88–94.

Kerr, T. (1987). Decentralized Filtering and Redundancy Management for Multisensor Navigation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-23(1):83–119.

Mutambara, A. G. O. (1998). *Decentralized Estimation and Control for Multi-Sensor Systems*. CRC Press, Boca Raton.

Nikiforov, I. (2002). Integrity monitoring for multi-sensor integrated navigation systems. In *Proceedings of the 15th International Technical Meeting of The Satellite Division of The Institute of Navigation (ION GPS 2002)*, pages 579–590.

Zhai, Y., Joerger, M., and Pervan, B. (2018). Fault exclusion in multi-constellation global navigation satellite systems. *Journal of Navigation*, 71(6):1281–1298.

APPENDIX

A. Sensor specifications

Table 2: IMU specifications

Parameter	Value	Units
Velocity random walk	0.005	$m/s/\sqrt(s)$
Accelerometer bias time constant	3600	s
Accelerometer scale factor standard deviation	300	ppm
Accelerometer time constant bias standard deviation	0.01	$m/s/s$
Angular random walk	0.2	$deg/\sqrt(hr)$
Gyroscope bias time constant	3600	s
Gyroscope scale factor standard deviation	150	ppm
Gyroscope bias standard deviation	2	deg/hr

Table 3: Odometer specifications

Parameter	Value	Units
Scale factor	1.14×10^{-3}	ppm

Table 4: Barometer specifications

Parameter	Value	Units
Initial height offset standard deviation	2	m
In run stability standard deviation	0.02	m
Decorrelation time	100	s
Measurement standard deviation	3	m

Table 5: 3D Laser specifications

Parameter	Value	Units
Measurement time accuracy	1×10^{-3}	s
Maximum range	40	m
Horizontal field of view	6.283	rad
Vertical field of view	0.5236	rad

Table 6: 2D Laser specifications

Parameter	Value	Units
Measurement time accuracy	1×10^{-3}	s
Maximum range	80	m
Range standard deviation	0.01	m
Angular resolution standard deviation	0.01	rad

Table 7: GPS specifications

Parameter	**Value**	**Units**
Pseudorange standard deviation	1	m
Carrier phase standard deviation	0.03	m
Receiver clock bias standard deviation	500	m
Receiver clock drift standard deviation	0.01	m/s
Receiver clock drift time constant	100	s
Pseudorange bias time constant	10	s
Carrier phase bias time constant	50	s