Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Voting-based Fault Detection for Aircraft Position Measurements with Dissimilar Observations ⋆

Tamas Grof * **Peter Bauer** *

* *Systems and Control Laboratory, Institute for Computer Science and Control (SZTAKI), Budapest, Hungary (e-mail: bauer.peter@sztaki.hu).*

Abstract: In this article a fault detection algorithm for aircraft position measurements is proposed using redundant sensor information during landing scenarios. This work was developed in the framework of the VISION EU H2020 research project. The aircraft's position can be determined via instrumental landing system, GPS and camera measurements. Considering these three sources a two out of three voting logic can be developed. After transforming the measured data sets to a common format two different methods are constructed to execute voting. The first is simple and well known thresholding where the measured position values are compared pairwise and threshold violations registered. As dissimilar data noise strengths can make thresholding unreliable the second method proposed by the authors is supplemented with an additional statistical evaluation where the measurements undergo a two-sample Z-test. Both methods were evaluated off-line with Monte-Carlo computer simulation. The tests showed that the proposed statistical method outperforms the straightforward thresholding approach.

Keywords: Fault detection, Fault isolation, Sensor failures, Threshold logic

1. INTRODUCTION

Redundancy concepts are inevitable in aerospace applications (Sklaroff (1976)) as the required levels of safety can only be achieved by redundant systems. Nowadays there is extensive research in the application of visual information as a redundant source in aerospace such as Gibert et al. (2018), VISION (2016) and Watanabe et al. (2019). The latter targets to apply vision information as a third redundant source of position besides GPS and instrumental landing system (ILS) during aircraft landing. However, the redundant information sources provide increased safety only if they are properly handled (Sklaroff (1976); Lii et al. (2006); Hoseinnezhad and Bab-Hadiashar (2006)). Redundancy provides the possibility to filter out corrupted sensory information. Considering an N number of sensors which measure the same variable and supposing that among them M sensors operate sufficiently, the faulty sensor(s) can be ruled out by using various algorithms such as voting logics or statistical methods. In this article we consider voting logic with $N = 3$ sensor sources and so $M = 2$ constraint leading to two out of three (2 out of 3) voting logic.

⋆ This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. The research presented in this paper was funded by the Higher Education Institutional Excellence Program. The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 690811 and the Japan New Energy and Industrial Technology Development Organization under grant agreement No. 062600 as a part of the EU/Japan joint research project entitled 'Validation of Integrated Safety-enhanced Intelligent flight cONtrol (VISION)'.

There are several voting algorithms well applicable for similar sources such as binary values Wu et al. (2007); Balasubramanian et al. (2016) representing exact voting and dissimilar sources representing inexact voting as in Latif-Shabgahi (2011); Karimi et al. (2014). Our application case requires 3 input inexact voting and conversion of multi-source information into a common format. However, because of the dissimilar noise levels the threshold selection can be challenging (Lii et al. (2006); Hoseinnezhad and Bab-Hadiashar (2006)). The noise level of camera position is about one order of magnitude higher than the GPS and ILS which means that a compatible threshold for pairwise differences would easily mask errors of the GPS or ILS and lead to missed detections. One possible way to overcome this is the application of soft voting with properly tuned Fuzzy sets (Hoseinnezhad and Bab-Hadiashar (2006)) however, that method does not directly consider the information about signal noise levels. On the other hand application of statistical Z-test Leblanc (2003) can directly incorporate this information and so make threshold selection easier. So this article targets to provide a Z-test-based 2 out of 3 voting logic for dissimilar sensor sources and compare it to conventional thresholding method through the positioning of a landing aircraft.

The structure of the article is as follows: Section 2 introduces the applied simple thresholding and statistical test-based methods. Section 3 compares the two methods based-on Matlab simulated data and finally Section 4 concludes the paper.

2. TWO OUT OF THREE VOTING LOGIC SOLUTIONS

Based-on the GPS, ILS and camera measurements during final approach a 2 out of 3 voting logic can be developed to determine whether one of the position information sources has degraded performance. The basic assumption is that only one fault occurs at a time and a second fault can occur only several seconds after it. The signals are pairwise compared to their respective error thresholds. As noise spikes can cause false alarms up-down counters are applied to make a certain decision about the degraded system. If one of the sources is voted out then with the remaining two systems the possibility of complete system degradation is monitored as having pairwise errors larger then the thresholds means that one of the systems is degraded but there is no possibility to decide which one. However, making an alarm upon this event can provide the opportunity to abort the position measurement-based automated landing.

In order to compare the different outputs of the GPS, ILS and camera systems some common parameters are needed. Both the GPS and camera provide position in a runway relative coordinate system, while the ILS measures the deviation from the glideslope and localizer references. Finally, the applied common parameters are the values provided by the ILS. Therefore, outputs of the other systems have to be transformed by using the following formulas:

$$\begin{aligned} dY &= -X_{east} \\ dZ &= -X_{down} - (-X_{north} + t_{td}) \cdot tan\gamma \end{aligned} \tag{1}$$

Where X_{north}, X_{east} and X_{down} refer to the position data from the GPS and camera in the runway relative coordinate system, γ is the glide slope angle while t_{td} is the distance of the touchdown point from the runway threshold. dY is the localizer deviation and dZ is the glide deviation (both considered in meter unit).

2.1 Simple thresholding method (THS)

The easiest way for pairwise data comparison is simple thresholding where the task is to find a nominal noise threshold for each system (ILS, GPS, camera) and generate the pairwise threshold values from this. As the ILS and camera estimation errors are distance dependent the nominal noise thresholds should also be.

Throughout the paper the following sensor pairings are considered and the pairwise thresholds generated by simple addition of the nominal ones:

(1) GPS-ILS pair (GvsI)
(2) Camera-GPS pair (CvsG)
(3) ILS-Camera pair (IvsC)

Threshold violation of a pair means that there is a fault in one of the systems. However, one threshold violation can be caused by a sudden spike in the data so it is important to prevent false alarms based-on this phenomenon. The solution is to apply up-down counters and a separate threshold for them. Upon violation of the counter threshold the fault can be detected. As the glide (gld) and localizer (loc) positions are handled separately six up-down counter values are defined as GPS_{gld}, ILS_{gld}, CAM_{gld}, GPS_{loc}, ILS_{loc} and lastly CAM_{loc}.

The rules of pairwise sub-system output comparison (separately for glide and localizer) are as follows:

(1) If all three pairings stay below the thresholds all of the count values are decreased by the *downcount* parameter if they are non-zero.
(2) If only one pair exceeds the given threshold there is no way to determine which one of the two sub-systems is defective so the count values dont change, except the third one is decreased by the *downcount* parameter.
(3) If two pairs exceed the given threshold the system which is featured in both can be considered the one causing the error. In that case the algorithm increases the count of that system by an *upcount* parameter while decreases the other two.
(4) As a worst case scenario, if all the pairings exceed their thresholds, all three count values will be increased.

The performance of the developed algorithm can be fine tuned by changing the *upcount* and *downcount* values and also by changing the counter decision threshold. After one of the systems was voted out, the monitoring of the remaining system pair continues to detect complete system failure when occurs.

The described algorithm has a weakness if the three sensor systems have very different noise levels. The pairwise thresholds of the pairs are equivalent to the sums of the noise thresholds. Therefore, in cases where a system has much higher noise level then the others, this can lead to the case that the pairwise threshold masks the errors in the low noise system and can lead to missed detections. One possible solution is the application of soft voting with fuzzy sets as in Hoseinnezhad and Bab-Hadiashar (2006). This article presents another solution, which is based-on statistical testing of the signals in the next part.

2.2 Statistical test-based method (STAT)

In order to counter the above mentioned problem a different fault detection algorithm was created by the authors which is based on statistical hypothesis testing. In this solution the sensor measurements are also transformed into glide and localizer deviations as common parameters. Instead of comparing the momentary information from the sensors improved results can be achieved through accumulating the data for a short time interval and then using the mean values of these data sets as the base for comparison. Comparing the accumulated data sets with statistical testing would provide more accurate fault detection. As the sample size increases the accuracy of the statistical approach grows as well but the system initialization time (until the first data set is gathered) also increases. Note that after gathering the first full data set, a moving-window technique can be used by adding the latest and removing the earliest data.

As it was mentioned, the base of the new method is a statistical hypothesis test. Since the sample size can be considerably large, the use of a two sample Z-test (see Leblanc (2003)) was selected (note that, if the size of data sets is modified it is advised to use a different test such as T-test for smaller sample sizes). The prerequisite for using the two sample Z-test is that the two samples must come from two independent but normally distributed data sets called populations. The central limit theorem states that: given a population with a finite mean μ and a finite non-zero variance σ^2, the sampling distribution of the mean approaches a normal distribution with a mean of μ and a variance of $\frac{\sigma^2}{n}$ as n - the sample size - increases. Therefore, with large sample sizes the sensory data means can be considered as normally distributed data sets. Since the measurements are derived from different sensors, it can be assumed that the samples are from different populations.

The statistical test examines whether the samples could come from populations where the difference of the means is Δ. In the given application the theoretical Δ value is zero as the same positions should be measured by all sensors. However, it is advisable to consider a position measurement tolerance both in glide and localizer considering nonzero $\Delta_{gld} = 1\%$ and $\Delta_{loc} = 1m$. This shows that in the localizer channel there is a constant threshold while the glide is distance dependent giving smaller altitude tolerance near to the threshold line. The Z-test returns with a decision about the difference of the population means. In order to carry out the two-sample Z-test the first step is to calculate the z-score using the following formula:

$$z = \frac{(\overline{x}_1 - \overline{x}_2) - \Delta}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \quad (2)$$

Where $\overline{x}_1$ and $\overline{x}_2$ are the mean values of the data sets, σ_1^2 and σ_2^2 are the variances of the data sets and finally n_1 and n_2 are the sample sizes ($n_1 = n_2 = n$ holds in this article). The equation stated above gives a z value as a result, which is the base for the Z-test. Statistical testing methods consider a null hypothesis, which is assumed to be initially true. In this case, it is assumed that samples are from populations where the difference of the means is Δ. The z value shows that how much the difference of sample means differs from Δ. The z value is a multiple of σ meaning that it shows the difference of means in terms of standard deviation. The end result of the test is a p-value which shows the probability of getting the actual result from populations where the difference of the means is Δ. The standard prespecified significance level is $\alpha = 0.05$. If the p value is smaller than the significance level the null hypothesis is rejected, otherwise it is assumed true. As mentioned, the Z-test returns with the decision whether the samples could come from certain populations where the difference of means is Δ. However, the Δ limit was chosen as the maximum value for the acceptable deviation between two sensor measurements. In order to fully carry out the fault detection, the two-sample Z-test should be performed for each value between 0 and Δ as one should ensure that the difference of means is not greater than Δ. However, it would require considerable computing power to perform the algorithm for every possible value hence it was decided that the statistical method should be supplemented with an additional step. Before applying the two-sample Z-test a simple comparison is carried out, which examines the difference of means of the data sets. If that difference is greater than Δ the statistical method with Δ parameter is carried out, otherwise the null hypothesis is accepted without further calculation. After the decision is made about the null hypothesis for each system pairings the 2 out of 3 voting logic is applied to detect the faulty sensor the same way (with up-down counters and related threshold) as in the case of the simple thresholding method.

3. COMPARISON OF METHODS

In this section the comparison of the two methods is presented. In order to fully analyse and compare the developed algorithms a Monte Carlo simulation test campaign was done in Matlab Software-in-the-loop (SIL) K-50 aircraft simulation (see Fig. 1).

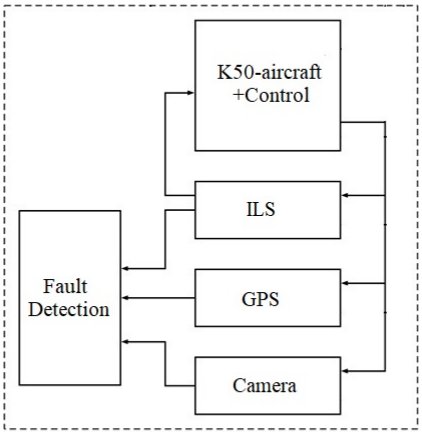

Fig. 1. Overview of the simulation containing the K-50 aircraft model and sensor models

Fig. 2. K-50 test aircraft (horizontal tail removed)

In the simulation, the aircraft (see Fig. 2) was guided during an approach maneuver with uncorrupted ILS data while the fault detection part was fed with possibly corrupted ILS, GPS and camera data.

In case of the ILS system the corrupted measurements being used in the simulation come from measurements of real landing scenarios (by Electronic Navigation Research Institute, Japan (ENRI)). The model assumes an object placed near the runway which causes perturbation in the ILS system. The ILS fault generation algorithm has two modes in total in accordance of the position of the interfering object. The simulated error values depend on the distance between the aircraft and the runway's threshold.

The simulated GPS unit again from ENRI has six simulation modes in total, ranging from a simulated SBAS system to large range error models. Other modes include the same SBAS system operating with four satellites, SBAS with ionospheric perturbation and three GPS modes such as regular, four satellite and large range error modes. The SBAS mode is considered the most accurate amongst them. Therefore, it was chosen as the nominal mode.

The simulated pinhole camera system adds distance dependent noise to runway relative position. The noise intensity decreases as the aircraft approaches the runway. This is because the relative effect of camera pixel noise decreases as the image plane object (runway) sizes increases. The noise intensity levels were obtained from Monte-Carlo simulation of an auto landing scenario considering image noise with 2 pixels variance of the onboard camera. The distance dependent noise is considered nominal as it can not be removed from the measurements. The random camera system error is the fixed orientation error of the camera, which comes from loose mounting or mounting error. It is considered to be time invariant during an approach for simplicity.

The distance dependent noise levels of the nominal working modes of the sensor systems were selected as noise thresholds. The pairwise thresholds were easily generated from them. The Z-test also uses the variance values of these nominal noises. Note that, in real flight scenarios the GPS and camera data frequency can be much lower than ILS frequency. However, the goal of this article is to compare the ideal performances of the two methods. Therefore, the sampling frequency of all systems was considered to be 100Hz with an $n = 100$ samples moving window in the STAT method.

The initial conditions (aircraft glideslope and localizer initial position error), camera orientation errors and GPS and ILS error modes were changed in the different test cases of the Monte-Carlo simulation. 16 batch of test runs were completed, which resulted in overall 816 simulated test cases. From the 816 test cases 784 include at least one fault while 528 include double (32 cases are fault free to test for false alarms). Each batch of test runs include 51 simulations while the main difference between the different batches lies in the different starting position from where the simulated landing was initiated. The initial position of the aircraft was set between $-10m$ and $10m$ for the localizer deviation and $-14m$ and $10m$ for glide deviation. The simulated batches also include scenarios where the initial position errors are zero. Another crucial difference between the test batches is the magnitude of the Camera orientation error (from the loose mounting), which was varied between $0.1°$ and $0.22°$. These values were selected for the reason that those magnitudes cause detectable faults but also the runway stays in the camera's field of view meaning that the camera can still provide the required measurements. The simulated camera orientation errors included scenarios where the loose mounting only affected the Roll or Pitch or Yaw values or in some cases all of them. The last difference between the batches is the time when sensor error modes were initiated between 1 and 17 seconds. The 51 cases within each batch include every possible error mode initiation order. The primarily corrupted sensor can be the GPS, the ILS or the Camera. The same holds for the second sensor degradation as all possible error sequences were tested. It is important to note that the initiation times were set in a way that enabled the algorithm to detect the first fault before activating the following one. As an example, one batch simulated scenarios where the possible error initiation times were 1-4-7 seconds for ILS, GPS and Camera sensors respectively. Therefore, in the case considering ILS-Camera sensor malfunction, the ILS degradation started at 1 second and the Camera degradation occur at 7 seconds in runtime.

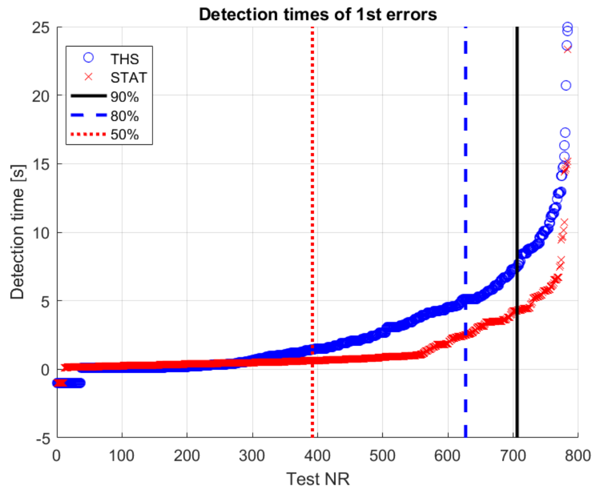

Fig. 3. Detection times of first faults with different methods

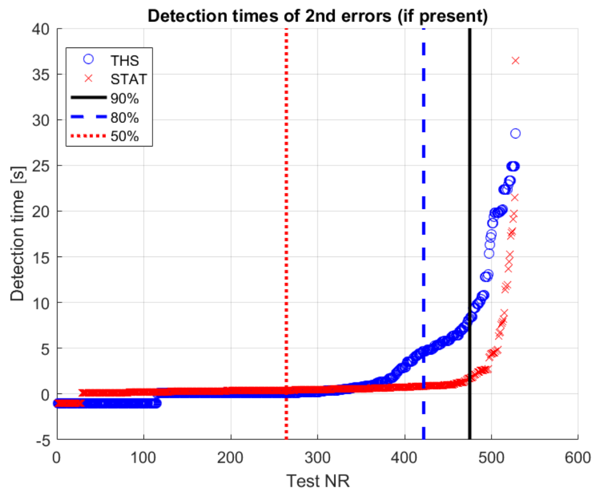

Fig. 4. Detection times of second faults with different methods

First, the fault detection times of the methods were compared calculated from error initiation of course. Fig.s 3 and 4 plot the detection times in ascending order with the different methods for first and second faults respectively. The number of test (Test NR) refers to the case number out of the overall 816 scenarios. The figures show that the threshold-based method gives lower detection times when it is able to detect the faulty sensor early after the error mode initialization. However, it gives higher detection times when the detection occurs later after error initialization. The statistical method is faster above detection times 0.44s for first fault and 0.55s for second, which represent most of the detection time ranges (around and above 50% of the test cases see Table 1).

Table 1. Detection times limits

Fault 1	50%	80%	90%
THS	1.44s	5.11s	7.46s
STAT	0.64s	2.49s	4.26s
Fault 2	**50%**	**80%**	**90%**
THS	0.11s	4.69s	8.28s
STAT	0.41s	0.83s	1.73s

Table 1 shows the maximum detection times for a given percent of the examined cases (The same test NR percent limits also shown in Fig.s 3 and 4 as vertical lines). In case of the first fault 90% of the cases is below 7.46s for the simple thresholding while below 4.26s for the statistical method. This is almost 43% reduction of detection time with the new method. For the 80% and 50% of test cases the reduction in statistical detection times is 51% and almost 56% respectively. So the new method gives faster detection reducing the detection time to about half of the other method's. Considering the second fault the detection time reductions are 79%, 82% and -273% for 90%, 80% and 50% of the data respectively. Note that the -273% deviation means only 0.3s absolute time which is more than acceptable (see Table 1). The results show that the simple thresholding method is slightly faster in detecting evident sensor corruptions meanwhile, the statistical method proved to be far more superior in detecting less noticeable faults.

Considering the missed detection (MD) and false alarm (FA) rates in Table 2, the missed detection rates are 61% and 77% better with the new method for first and second faults respectively with the price of an increase in the false alarm rate. However, false alarm rates below 2% can be acceptable even in real application.

Table 2. Missed detection and false alarm rates

Rate %	MD Fault 1	MD Fault 2	FA Fault 1	FA Fault 2
THS	5.4	15.7	0	0.25
STAT	2.08	3.55	0.25	1.72

Fig.s 6 and 5 show an ILS error case (for the glide subsystem) where with the statistical method the mean values of the GPS-ILS (GvsI) pair and ILS-Camera (IvsC) pair differences obviously violate the threshold and indicate ILS fault. Considering the simple thresholding method the violation of the GPS-ILS (GvsI) pair is obvious but for the ILS-Camera(IvsC) pair it is uncertain and there are also violations with Camera-GPS (CvsG) pair. This method finally detected a GPS error instead of the ILS so the uncertain thresholding resulted in missed detection and false alarm together which is very dangerous. This is caused by the masking effect of the camera noise which is almost an order of magnitude larger than the other system noises as the pairwise thresholds in Fig. 5 show.

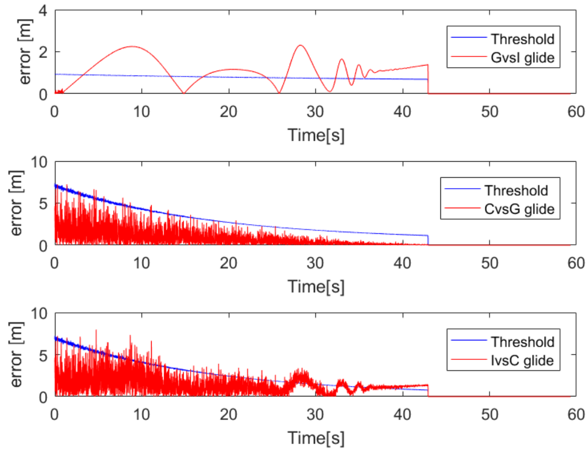

Fig. 5. Glide part error detection with simple thresholding method

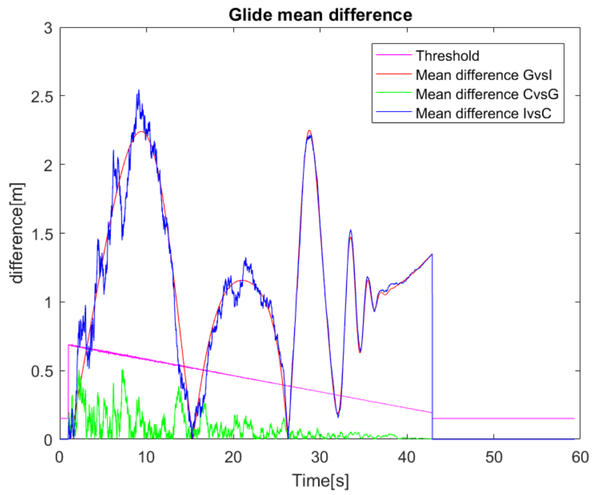

Fig. 6. Glide part error detection with statistical method

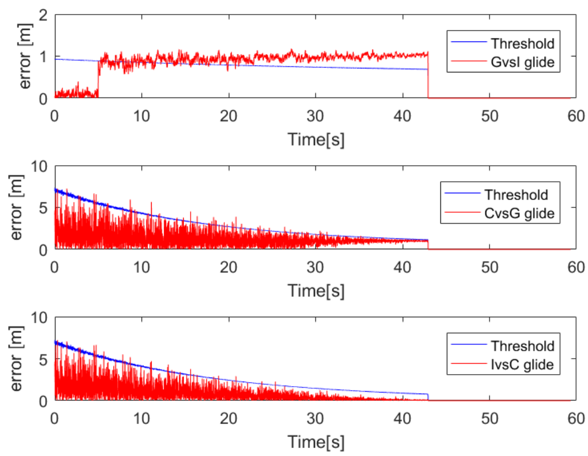

Fig. 7. Glide part error detection with simple thresholding method

Fig.s 7 and 8 show a case when both the thresholding and the statistical methods do correct detection but the thresholding is much slower (16.3s detection time instead of 1.15s) because spike violations of the threshold should lead to the final violation of counter threshold.

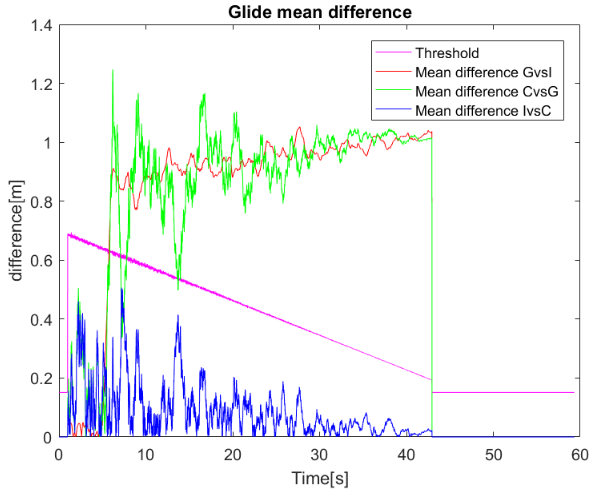

Fig. 8. Glide part error detection with statistical method

As a summary, it can be stated that the statistical method outperforms the simple thresholding method in terms of average detection times in both first and second sensor malfunctions. It also decreased the number of missed detections with the price of an increase in the false alarm rate. Another aspect of the new method is that the computational load is higher, since more operations are required to perform the Z-test. Likewise, more memory is needed to store the required data sets compared to the simple thresholding method however, these issues can be easily handled by current high performance on-board hardware architectures.

4. CONCLUSION

In this paper two algorithms are presented to provide fault detection of aircraft position measurements from redundant sensor information during final approach. Both of them is 2 out of 3 voting with simple thresholding in the first and statistical test-based decision in the second. The latter is the a new method proposed by the authors. As explained in the article dissimilar data noise intensity increases the possibility of missed detections in the first method as higher noise levels can mask small errors. To counter this effect the second method was proposed where the measurements undergo a two-sample moving window statistical Z-test, which examines whether the measured position values are similar or not from a statistical point of view. After the description of the two algorithms, a thorough comparison through Monte-Carlo simulation in Matlab/Simulink was carried out to test the functionality regarding average detection times, missed detection and false alarm rates. The simulation campaign consists of 816 simulation cases overall, which covers several of the possible scenarios. The comparison showed the superiority of the statistical method regarding detection times and missed detections with the price of slightly increased false alarm rate. Possible improvements can be acquired through tuning the voting logic parameters such as the counter thresholds or the parameters of the Z-test. Besides fine tuning future work plans to include application with real flight test data and the related limited sampling frequencies of GPS and camera.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the contribution of Yoko Watanabe and her team from ONERA by providing the K-50 simulation model, Ryota Mori and his team from ENRI by providing the GPS and ILS error models and Antal Hiba from SZTAKI by providing the Monte-Carlo simulation-based camera error model of the simulation.

REFERENCES

Balasubramanian, P., Prasad, K., and Mastorakis, N.E. (2016). A Fault Tolerance Improved Majority Voter for TMR System Architectures. *WSEAS Transactions on Circuits and Systems*, 15. URL `http://arxiv.org/abs/1605.03771`.

Gibert, V., Plestan, F., Burlion, L., Boada-Bauxell, J., and Chriette, A. (2018). Visual estimation of deviations for the civil aircraft landing. *Control Engineering Practice*, 75, 17 – 25. doi: https://doi.org/10.1016/j.conengprac.2018.03.004.

Hoseinnezhad, R. and Bab-Hadiashar, A. (2006). Fusion of Redundant Information in Brake-By-Wire Systems Using a Fuzzy Voter. *JOURNAL OF ADVANCES IN INFORMATION FUSION*, 1(1).

Karimi, A., Zarafshan, F., Al-Haddad, S.A.R., and Ramli, A. (2014). A Novel N -Input Voting Algorithm for X -by-Wire Fault-Tolerant Systems. *The Scientific World Journal*, 2014, 672832. doi:10.1155/2014/672832.

Latif-Shabgahi, S. (2011). An Integrated Voting Algorithm for Fault Tolerant Systems. In *In Proc. of 2011 International Conference on Software and Computer Applications*.

Leblanc, D.C. (2003). *Statistics: Concepts and Applications for Science*. Jones & Bartlett Publishing.

Lii, N.Y., Sturm, S., and Coombs, T.A. (2006). A Study on Sensor Fusion for Fault Tolerant Brake-by-Wire Driver Input Design with Dissimilar Sensors. In *In Proc. of SAE 2006 World Congress & Exhibition*.

Sklaroff, J.R. (1976). Redundancy Management Technique for Space Shuttle Computers. *IBM Journal of Research and Development*, 20(1), 20–28. doi: 10.1147/rd.201.0020.

VISION (2016). Vision project webpage. URL `https://w3.onera.fr/h2020_vision/node/1`.

Watanabe, Y., Manecy, A., Hiba, A., Nagai, S., and Aoki, S. (2019). Vision-integrated navigation system for aircraft final approach in case of GNSS/SBAS or ILS failures. AIAA SciTech Forum. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2019-0113. URL `https://doi.org/10.2514/6.2019-0113`.

Wu, J.Y., Duh, D.R., Wang, T.Y., and Chang, L.Y. (2007). Fast and Simple On-Line Sensor Fault Detection Scheme for Wireless Sensor Networks. In T.W. Kuo, E. Sha, M. Guo, L.T. Yang, and Z. Shao (eds.), *Embedded and Ubiquitous Computing*, 444–455. Springer Berlin Heidelberg, Berlin, Heidelberg.