A Fault Tolerant Perception system for autonomous vehicles

Miguel Realpe[1,2], Boris X. Vintimilla[1], Ljubo Vlacic[2]

1. Escuela Superior Politécnica del Litoral, ESPOL, CIDIS - FIEC, Campus Gustavo Galindo Km 30.5 Vía Perimetral, Guayaquil, Ecuador
E-mail: [mrealpe, bvintim]@fiec.espol.edu.ec

2. Intelligent Control Systems Laboratory, Griffith University. Brisbane, Australia
E-mail: l.vlacic@griffith.edu.au

Abstract: Road driving environments are complex, unstructured and highly changeable. A safe driving is, thus, becoming quite challenging task, in particular from the view point of development and deployment of autonomous vehicles-based urban transport systems. In that context, the reliable perception appears as a one of the main enabling strategies in developing safe autonomous driving. Currently, many autonomous vehicles are being tested on public roads with the objective of demonstrating the capability of operating in real world situations. A big effort has been focused towards creating fault-free autonomous vehicles. Nevertheless, fault tolerant perception for autonomous vehicles still needs to be further developed in order to create autonomous vehicles capable of driving under real road traffic conditions since on-board vehicle sensors may fail due to bad calibration, erroneous readings, physical or electrical failures, etc. A multi-sensor based vehicle architecture is a logical response to this issue. While the multi-sensor concept often relates to the strategy of using a variety of sensor types, this research has been focused on to the case when all sensors are vision sensors, either identical or different from each other. This paper proposes a Fault Tolerant Perception paradigm that deals with possible sensor faults by defining the Federated Data Fusion Architecture designed to detect a faulty sensor and reduce its impact on to the safe autonomous driving. The proposed architecture minimises the influence of faulty data allowing the system to enter in a tolerated error state, where a recovery action can be performed to avoid failures. The developed architecture was then adapted towards meeting requirements of the KITTI Vision Benchmark Suite. Experimental results demonstrated the feasibility of the developed fault tolerant perception paradigm to successfully detect early faulty data from a singular sensor and to minimise the influence of that faulty sensor in the fusion process.

Key Words: Fault Tolerant Perception, Sensor Data Fusion, Fault Tolerance, Autonomous Vehicles, Federated Architecture.

1 Introduction

The dependability of a system is the ability to perform complying time-related quality characteristics according to its specification when required. It is used as a collective term that includes availability, reliability, recoverability, maintainability, and maintenance support performance [1]. A threat is any event that negatively affects the dependability. It can be classified as faults, errors, or failures. A fault is active when it produces an error, otherwise it is dormant; a failure occurs if an error is not detected, resulting in an output that is inconsistent with the system specification.

Causes of a fault can be either: software imperfections due to software aging or development mistakes; or hardware defects, such as production error, internal deterioration or external physical interference (electromagnetic, radiation, etc.). With respect to their appearance faults can be classified as hard and soft faults. Hard faults are presented in a stepwise form when data changes abruptly from its normal state to a faulty one, while soft faults are slow degradations in the data. As a consequence of active faults, the content or the timing of the output may deviates from implementing the system function, creating a failure.

The period of delivery of a failed system is called service outage, whereas the transition back to the correct system is a service restoration. In order to avoid service outage and to reduce service restoration to the minimum, dependability manages four different approaches:

- Fault prevention. Means to prevent the incorporation of faults into the system. It is focused on good implementation techniques and development methodologies such as adaptability, modularity and interoperability.
- Fault removal. Denotes the reduction of the number and severity of faults by applying preventive maintenance or by verifying that the system satisfies requirement properties (e.g. static analysis, model checking, testing) and then implementing corrections.
- Fault forecasting. Means to estimate the present number of faults and their likely consequences. It usually applies to faults that cannot or should not be removed [2]; instead their influence is qualitatively or quantitatively assessed.
- Fault tolerance. Denotes the capability of a system to avoid failures in the presence of faults.

Fault prevention and fault removal try to avoid the presence of faults. However, in complex systems it is difficult to achieve their fault-free operations. On the other hand, fault tolerance and fault forecasting strategies both embrace the existence of faults and are focused on keeping the systems operational under the presence of one or more faults. The focus of the work presented in this paper relies on the latter case, specifically on fault tolerance, since very little research has been done in that area for perception systems of autonomous vehicles.

[*]This work is supported by the National Secretary of Superior Education, Science, Technology & Innovation of Ecuador (SENESCYT) and the Escuela Superior Politecnica del Litoral.

This paper is organized as follows: In section 2 the models for sensor data fusion are explored. Then, the fault tolerance concepts and architectures are reviewed in section 3. Section 4 explains the proposed fault tolerant perception system. Experimental results are shown in section 5. Finally, Section 6 concludes this paper.

2 Sensor Data Fusion

Autonomous vehicles require complete and accurate information about their environment to support their operation, but this information cannot be achieved using any one single sensor [3]. Consequently, a collection of sensors must be used and strategies to combine all sensory data have to be defined. Data fusion is the group of techniques used for combining data from multiple sensors (either of the same or different types) and related information into a common representational format to achieve more accurate inferences than could be achieved by using a single, independent sensor [4, 5].

Data fusion not only adds a statistical advantage by adding N independent observations, but also improves observability resulting in reduced error regions [5].

The most cited architecture for data fusion is the JDL model that was created by the American Joint Directors of Laboratories Data Fusion Subpanel [6, 7]. The JDL model has a database and different levels of data processing interconnected by a common bus. The core of the JDL model refers to Levels 1-3. Level 4 (process refinement) is related to resource management and is not part of the core levels, but it can be used concurrently for giving a fault tolerant framework to the system. The JDL model does not request the levels to be processed in a sequential order and they can be executed concurrently. However, data fusion system designers have consistently assumed an ordering on the JDL levels [8]. A fifth level was proposed by Blasch and Plano [9] to incorporate the user in the fusion process.

Polychronopoulos and Amditis [10] proposed the ProFusion2 model (PF2), a revision of the JDL model, in order to apply it in multi sensor automotive safety systems. They grouped JDL levels into layers to add hierarchical structure. Also, they established inter-level and within-layer interactions, excluding the process refinement from the original JDL model. The ProFusion2 model (PF2) is described by 3 layers: I, Perception; II, Decision/ Application and III, Action/Human Machine Interface (HMI). The processing of the layers is done in a hierarchical structure from the lower to the upper level.

3 Fault Tolerance

The concept of fault tolerant systems deals with the problem of ensuring correct system service in the presence of faults in order to avoid unplanned behaviours [11]. The implementation of fault tolerance in general implies three steps: error detection, error location, and recovery.

The most common sensor fault detection methods for complex systems are the analytical methods, also called model-based methods. These methods require a mathematical model of the system along with available inputs and outputs of the system in order to produce a group of features. Then, these features are compared with the system variables, creating residual values. The residuals that differ from the nominal values are called symptoms and can be subject to a symptom-fault classification in order to detect a fault and its location (fault diagnosis) [12]. Fault diagnosis is based on observed symptoms and experience-based knowledge of the system. One approach is the use of classification methods, where the relation between symptoms and faults are determined experimentally in a previous phase of the system. Another approach is the use of inference methods, where causal relations are created in the form of rules based on partially known relationships between faults and symptoms.

In general, model based methods are categorized as observer-based [13, 14]. parity equations [15, 16] and parameter estimation techniques [17].These methods are very popular for fault tolerant control systems because models of control systems can usually be obtained. Nevertheless, soft computing techniques, such as neural networks, evolutionary algorithms and support vector machines (SVM), are being developed for fault detection and diagnosis (FDD) for other applications, because obtaining a good model of the system is a complex task and it is not always possible [18].

In recent years, only a few specific solutions of fault tolerant perception systems for autonomous vehicles have been developed. However, many researchers have implemented fault tolerant modules for autonomous vehicles in areas such as vehicle navigation sensors. The architectures applied for navigation systems can be grouped into centralized architectures and federated architectures.

Centralized architecture uses a global filter that processes the measurements of all local sensors. The main advantage of this architecture relies on obtaining the most accurate solution as a consequence of the minimal information loss. Furthermore, hard sensor failures can be detected using residual tests by contrasting each sensor measurement with the accumulated estimation of the global filter [19]. However, the centralized filter is not robust against soft sensor faults [20]. Using a single filter tends to make all data agree. Therefore, if there is an undetected soft fault, such as a bias shift sensor, it will result in a displacement of the global estimate toward the faulty data. As a consequence, a good sensor might be declared faulty for disagreeing with the global estimate. In addition, once the soft fault is large enough to become detectable, it has already contaminated the data of the other sensors.

Federated architecture [19] is composed of a group of local filters, that operate in parallel, and a master filter. It tries to isolate faulty sensors before their data becomes integrated into the system. For that, each local filter fuses data independently from a specific sensor with common information from a reference sensor. Then, the local estimates are fused in the master filter in order to generate the best global estimation.

Local filters may use information from the master filter by setting parameters applying a feedback signal in order to improve their performance. However, if the master filter admits faulty data from any local filter, this will affect the whole system through the feedback signals. On the other hand, federated filter architecture with no feedback is highly fault tolerant and provides a good response to soft faults. Since each local filter is independent, data from faulty sensors can be easily isolated before being integrated with the entire system, while the remaining sensors immediately generate the new federated filter system. In [21] a no-feedback federated filter is implemented for the multi-sensor navigation system of an unmanned aerial vehicle using a INS sensor as reference in order to obtain a good fault-tolerant performance.

After identifying faults, a reconfiguration of the system architecture is required. Fault recovery can be achieved using direct redundancy or analytical redundancy [22]. With direct redundancy, a spare module is employed to replace the faulty one. On the other hand, analytical redundancy implies utilizing the working modules to complete the tasks which failed. For instance, if there is a fault in a laser scanner of an autonomous vehicle, the information from two cameras can be used instead to create range data and compensate for the laser scanner functions.

The effects of a fault that transits from dormant to error and then to failure, in a fault tolerant system can be modelled by a sequence of states (see Fig. 1). Initially the system executes in a correct state (C). The transition from a state C to a tolerated error state (T) is constrained by a fault action. In the state T, a recovery action returns the system to a state C, whereas correct or fault actions maintain the same state. However, if the fault tolerant system has a limitation that is overrun by the propagation of a fault action to the outputs, the system will produce a failure. Thus, when the system is running in a tolerated error state a recovery action according to the fault action should be executed (e.g. resetting for software aging, synchronization for timing error, etc).

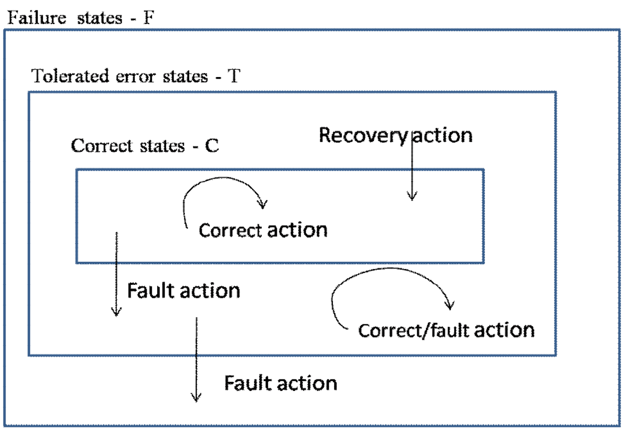

Fig. 1: Fault Tolerant System state space

4 Proposed Fault Tolerant Perception

The common modules used by autonomous vehicles' perception systems are related to localization, road detection and obstacle detection. The study presented in this paper is limited to object detection for medium range obstacles and it is focused on fault tolerance. Thus, the causes of a fault are not important and just the faulty data itself is relevant because it is used to obtain the residuals in order to find an active fault.

4.1 General model

The general model proposed for information fusion is a modification of the PF2 model; we suggest integrating the process refinement into the sensor fusion process. Furthermore, the process refinement may interact with all the layers, allowing the implementation of fault tolerance not only in the perception layer, but for the entire autonomous vehicle system (see Fig. 2).

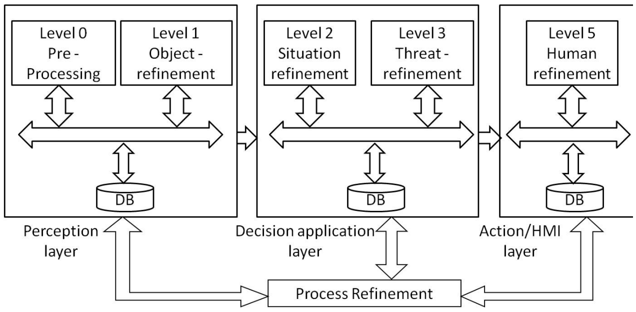

Fig. 2: Data fusion model for fault tolerant implementation [23]

A federated perception architecture is proposed to fuse sensor data into a fault tolerant framework (see Fig. 3). The proposed architecture has been divided into different modules: one object detection for each sensor type, one local fusion for each support sensor, one master fusion and the FDD module. The federated perception architecture uses a master fusion module as a reference and various local fusion filters to validate information. The purpose is to use a highly reliable sensor as a reference sensor. Also, M and N sensors of different categories are proposed. Where the sum of M and N is at least two, so the FDD module would have at least three discrepancies group values to compare and find a fault.

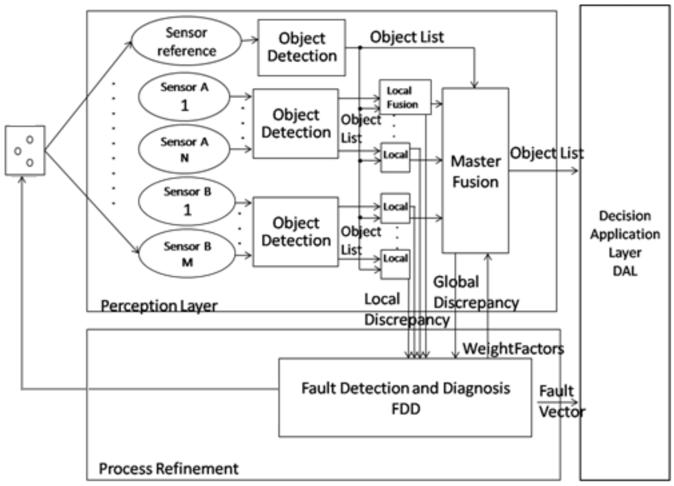

Fig. 3: General view of the proposed fault tolerant perception architecture

The sensor fusion performs with detected objects data in order to increase tracking consistency and to increase the adaptability to new types of sensors, which can be added by implementing the corresponding object detection module. Each local filter individually processes the compatible data between the reference sensor and the data provided by each sensor. The master fusion module combines information of local filters using their lists of objects. In addition, it realizes a tracking of the objects. The discrepancies from local fusion modules and the master fusion are used by the FDD to estimate the residuals values in a support vector machine (SVM) in order to identify if an error occurs in the sensors.

4.2 Implementation

The KITTI dataset [24-26] has been chosen in order to implement the proposed system. It includes information from greyscale and colour stereo sequences and 3D Velodyne point clouds synchronized at 10Hz. Fig. 4 shows the proposed fault tolerant perception system adapted to the data available from the KITTI dataset. The Velodyne is used as the main sensor in order to take advantage of the high reliability and amount of information provided, while the vision sensors are used to complement the information.

Object Detection (OD) combines diverse simple and fast state-of-the-art detection algorithms in order to obtain cues of possible obstacles in a real-time frame. There is one distinct OD category for each sensor type (Velodyne, vision sensor) and one instance of the respective category is created for each physical sensor (OD Velodyne, OD vision sensor1, OD vision sensor2).

LF creates a single objects list using data from a specific sensor and the reference sensor. In addition, it creates the discrepancy values between those sensors. The discrepancies values are features that latter are compared in the FDD module to create residuals that evaluate the presence of faults. More details of the implementation of OD and LF can be found in [23].

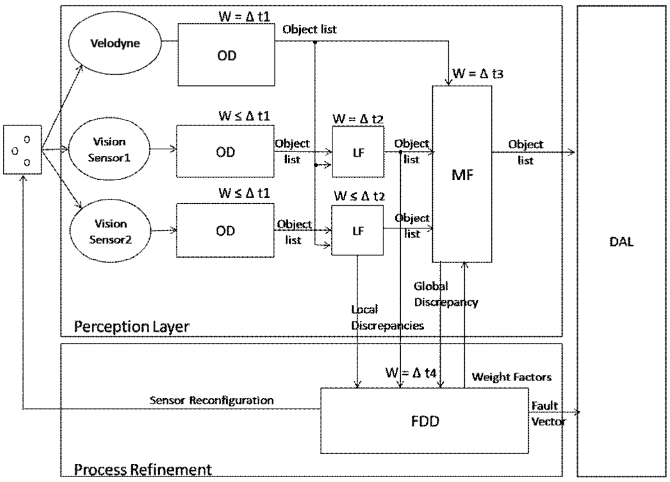

Fig. 4. Fault tolerant perception system for KITTI dataset [27].

[27] describes MF. It combines data from the reference sensor and the LF modules. It uses SVM to identify patterns between the objects and the weight of each sensor in order to validate pixels in the objects. In addition, multiple object tracking is performed in order to provide information about the future position of objects.

FDD applies SVM to recognize changes in the discrepancies from MF and LF modules. It creates a residual value which is mapped to 0 or 1, representing the presence or absent of a fault. In order to avoid false positives the output from the SVM is consider only if a faulty response is given after N consecutive outputs. Then, the respective sensor is reconfigured to a lower priority (high->low->off) [28].

Since the nature of the proposed vision based OD algorithm is focused on mobile obstacles, the discrepancies values are grouped into static and dynamic features in order to avoid errors introduced by dynamic objects showing high discrepancies produced between vision OD and Velodyne OD. Thus, the SVM model in FDD is trained with a vector of 18 features, which represent 6 discrepancies values (3 static and 3 dynamic) from each of the 3 fusion modules (2 locals and 1 master).

Fig. 5 [27] shows the states of the system when a fault is simulated. The transition actions are constrained by the outputs of FDD. The limitation of the current fault tolerant perception system is given by the number of faulty sensors, which is T/2 -1, where T represents the total number of sensors. For the KITTI dataset with the current configuration it means that 1 of the 3 sensor can be faulty at the time.

5 Experimental Results

The proposed architecture was tested using a sequence of 161 images from the KITTI dataset in a Core i5 CPU at 3.10 GHz. The frequency of the sensor acquisition was 10 Hz, which produces cycles of 100 ms. The maximum processing time of the system per cycle was 70 ms, which maintains the execution time of the system inside the frame given by the frequency of the sensor acquisition. The SVM models were trained using a subset of 25 representative images. The FDD model was trained using 635 vectors, whereas the MF model was trained with 505620 vectors.

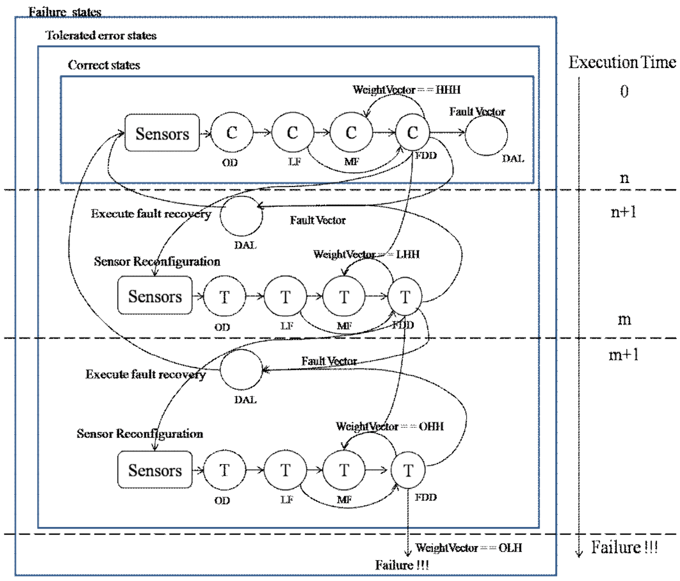

Fig. 5. System fault state space [27].

For the soft fault in the vision sensor 1, a displacement to the right by 30 cm in the objects was introduced. Likewise, a displacement to the left by 30 cm in the objects from the Velodyne sensor was introduced for the second experiment.

Then system was run in a correct state with a weight value of high for all the sensors. Eventually, FDD changes the weight for the faulty sensor from high to low and then off. Fig. 6 shows the output of the FDD module for the soft fault in vision sensor 1 (blue) and for the soft fault in the Velodyne sensor (red), when the respective sensor was reconfigured to a lower priority every time that the SVM response resulted positive for 4 consecutive images.

In a previous work [28] it was found that the effect of the sensor weight on MF can be appreciated in the reduction of false positives, while the number of detected objects and the number of pixels positively classified as part of an object are similar for all the cases (with and without fault). Fig 7 shows the percentage of pixels from the detected dynamic objects that are false positives. The red line represents the case of the system executing without fault tolerance, this means that sensor weights were not changed, despites the values of the FDD outputs. On the other hand, the blue line depicts the case of the fault tolerant system, where a tolerated error state was activated accordingly with the FDD outputs.

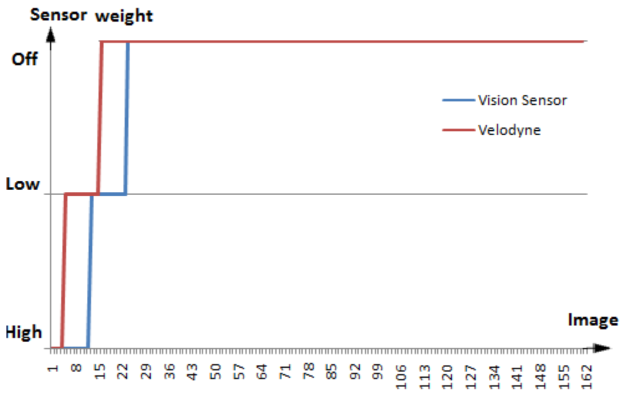

Fig. 6. Fault output from FDD.

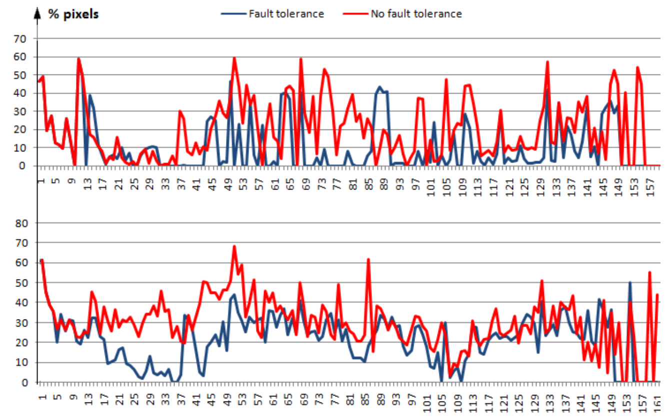

Fig. 7. False positives detection of MF with faulty data from top) Sensor vision, bottom) Velodyne.

For the case of faulty data from sensor vision the false positives pixels are reduced up to 59% with an average of 8% when fault tolerance is implemented. Similarly, false positives are reduced up to 47% with an average of 8% when fault tolerance is implemented in the presence of a fault in the Velodyne sensor.

6 Conclusions

A federated sensor data fusion architecture is proposed in order to provide fault tolerance to autonomous vehicle's perception system. The proposed architecture minimizes the influence of faulty data allowing the system to enter in a tolerated error state, where a recovery action can be performed to avoid failures. It integrates the process refinement to the fusion process by combining data from sensors with the sensors weight feedback provided in real-time by the Fault Detection and Diagnosis module.

The system has successfully detected early faulty data from a singular sensor and it has minimized the influence of the faulty sensor up to 59% with an average of 8%.

References

[1] International Electrotechnical Commission. (2016). *Electropedia del 192 Dependability, 192-01-22 Dependability*. Available: http://www.electropedia.org,

[2] M. Dacier, "A Fault Forecasting Approach for Operational Security Monitoring," in *Dependable Computing for Critical Applications 4*. vol. 9, F. Cristian, *et al.*, Eds., ed: Springer Vienna, 1995, pp. 215-217.

[3] C. Urmson, *et al.*, "Autonomous driving in urban environments: Boss and the Urban Challenge," *Journal of Field Robotics*, vol. 25, pp. 425-466, 2008.

[4] H. B. Mitchell and Knovel (Firm), *Multi-sensor data fusion: an introduction*. Berlin ; New York: Springer Verlag, 2007.

[5] D. L. Hall, *et al.*, *Handbook of multisensor data fusion*. Boca Raton, FL: CRC Press, 2001.

[6] F. E. White, "Data Fusion Lexicon " JOINT DIRECTORS OF LABS WASHINGTON DC. 1991.

[7] A. N. Steinberg, *et al.*, "Revisions to the JDL data fusion model," *Sensor Fusion: Architectures, Algorithms, and Applications III*, vol. 3719, pp. 430--441, 1999.

[8] J. Raol, *Multi-Sensor Data Fusion with Matlab*: CRC Press, Taylor & Francis Group, 2009.

[9] E. P. Blasch and S. Plano, "Level 5:User Refinement to Aid the Fusion Process," B. V. Dasarathy, Ed., ed, 2003.

[10] A. Polychronopoulos and A. Amditis, "Revisiting JDL model for automotive safety applications: the PF2 functional model," in *Information Fusion, 2006 9th International Conference on*, 2006, pp. 1-7.

[11] R. Isermann, *Fault-Diagnosis Applications: Model-Based Condition Monitoring: Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems*: Springer Berlin Heidelberg, 2011.

[12] R. Isermann. (2011). *Fault-diagnosis applications model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Available: http://dx.doi.org/10.1007/978-3-642-12767-0

[13] M. Hilbert, *et al.*, "Observer Based Condition Monitoring of the Generator Temperature Integrated in the Wind Turbine Controller," *EWEA 2013 Scientific Proceedings : Vienna, 4 -7 February 2013*, pp. 189-193, 2013.

[14] G. Heredia and A. Ollero, "Sensor fault detection in small autonomous helicopters using observer/Kalman filter identification," in

Mechatronics, 2009. ICM 2009. IEEE International Conference on, 2009, pp. 1-6.

[15] M. Muenchhof, "Comparison of change detection methods for a residual of a hydraulic servo-axis," pp. 1854-1854, 2005.

[16] C. W. Chan, *et al.*, "Application of Fully Decoupled Parity Equation in Fault Detection and Identification of DC Motors," *Industrial Electronics, IEEE Transactions on*, vol. 53, pp. 1277-1284, 2006.

[17] T. Escobet and L. Trave-Massuyes, " Parameter estimation methods for fault detection and isolation," *Bridge Workshop Notes,* 2001.

[18] N. Meskin and K. Khorasani, *Fault detection and isolation : multi-vehicle unmanned systems*. New York: Springer, 2011.

[19] N. A. Carlson, "Federated filter for fault-tolerant integrated navigation systems," in *Position Location and Navigation Symposium, 1988. Record. Navigation into the 21st Century. IEEE PLANS '88., IEEE*, 1988, pp. 110-119.

[20] P. J. Lawrence, Jr. and M. P. Berarducci, "Comparison of federated and centralized Kalman filters with fault detection considerations," in *Position Location and Navigation Symposium, 1994., IEEE*, 1994, pp. 703-710.

[21] H.-m. Chen, *et al.*, "Research on scheme and algorithm of high-precision fault-tolerant integrated navigation for HALE UAV," in *Systems and Control in Aeronautics and Astronautics (ISSCAA), 2010 3rd International Symposium on*, 2010, pp. 768-773.

[22] H. A. Aldridge, "Robot position sensor fault tolerance," Ph.D. 9713717, Carnegie Mellon University, United States -- Pennsylvania, 1996.

[23] M. Realpe, *et al.*, "Towards fault tolerant perception for autonomous vehicles: Local fusion," in *Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), 2015 IEEE 7th International Conference on*, 2015, pp. 253-258.

[24] A. Geiger, *et al.*, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 3354-3361.

[25] A. Geiger, *et al.*, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research,* vol. 32, pp. 1231-1237, September 1, 2013 2013.

[26] J. Fritsch, *et al.*, "A new performance measure and evaluation benchmark for road detection algorithms," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, 2013, pp. 1693-1700.

[27] M. Realpe, *et al.*, "Multi-sensor Fusion Module in a Fault Tolerant Perception System for Autonomous Vehicles," presented at the 2016 2nd International Conference on Robotics and Artificial Intelligence, Los Angeles, USA, 2016.

[28] M. Realpe, *et al.*, "Sensor Fault Detection and Diagnosis for autonomous vehicles," *MATEC Web of Conferences*, vol. 30, p. 04003, 2015.