

Article

Real-time Vehicle Roll Angle Estimation based on IoT low-cost devices and Neural Networks

Javier García Guzmán ^{1,*}, Lisardo Prieto González ^{1,†}, Jonatan Pajares Redondo ^{2,†}, Mat Max Montalvo Martínez ^{1,†} and María Jesús L. Boada ^{2,†}.

¹ Computer Science and Engineering Department, Institute for Automotive Vehicle Safety (ISVA), Universidad Carlos III de Madrid, Avda. de la Universidad 30, 28911 Leganés, Madrid, Spain; lpgonzal@inf.uc3m.es (L.P.G.); mamontal@pa.uc3m.es (M.M.M.M.),

² Mechanical Engineering Department, Institute for Automotive Vehicle Safety (ISVA), Universidad Carlos III de Madrid, Avda. de la Universidad 30, 28911 Leganés, Madrid, Spain; jopajare@ing.uc3m.es (J.P.R.), mjboada@ing.uc3m.es (M.J.L.B.)

* Correspondence: jgarciag@inf.uc3m.es; Tel.: +34-91-624-9988

† These authors contributed equally to this work.

Abstract: Given the high number of vehicle-crash victims, it has been established as a priority to reduce this figure in the transportation sector. For this reason, many of the recent researches are focused on including control systems in existing vehicles, to improve their stability, comfort and handling. These systems need to know in every moment the behavior of the vehicle (state variables), among others, when the different maneuvers are performed, to actuate by means of the systems in the vehicle (brakes, steering, suspension) and, in this way, to achieve a good behavior. The main problem arises from the lack of ability to directly capture several required dynamic vehicle variables, such as roll angle, from low-cost sensors. Previous studies demonstrate that low-cost sensors can provide data in real-time with the required precision and reliability. Even more, other research works indicate that neural networks are efficient mechanisms to estimate roll angle. Nevertheless, it is necessary to assess that the fusion of data coming from low-cost devices and estimations provided by neural networks can fulfill the reliability and appropriateness requirements for using these technologies to improve overall safety in production vehicles. Because of the increasing of computing power, the reduction of consumption and electric devices size, along with the high variety of communication technologies and networking protocols using Internet have yield to Internet of Things (IoT) development. In order to address this issue, this study has two main goals: 1) Determine the appropriateness and performance of neural networks embedded in low-cost sensors kits to estimate roll angle required to evaluate rollover risk situations. 2) Compare the low-cost control unit devices (Intel Edison and Raspberry Pi 3 Model B), to provide the roll angle estimation with this artificial neural network-based approach. To fulfil these objectives an experimental environment has been set up composed of a van with two set of low-cost kits, one including a Raspberry Pi 3 Model B, low cost Inertial Measurement Unit (BNO055 - 37€) and GPS (Mtk3339 - 53€) and the other having an Intel Edison System on Chip linked to a SparkFun 9 Degrees of Freedom module. This experimental environment will be tested in different maneuvers for comparison purposes. Neural networks embedded in low-cost sensor kits provide roll angle estimations very approximated to real values. Even more, Intel Edison and Raspberry Pi 3 Model B have enough computing capabilities to successfully run roll angle estimation based on neural networks to determine rollover risks situation fulfilling real-time operation restrictions stated for this problem.

Keywords: Real-time estimation; IoT; Artificial Neural Network; Vehicle dynamics; Roll angle; low cost devices; Raspberry Pi 3 Model B, Intel Edison, FANN.

1. Introduction

The high rate of vehicle-crash victims has a fatal economic and social impact in today's societies. That is why nowadays road vehicles incorporate safety systems in order to reduce accidents. In particular, road crashes where heavy vehicles are involved cause more severe damage because they are prone to rollover. For this reason, many researches are focused on developing RSC (Roll Stability Control) systems.

Concerning the design of RSC systems with an adequate performance, it is mandatory to know the dynamics of the vehicle. One of the most important parameters related to rollover dynamics is the roll angle. The problem is that this angle cannot be measured directly using low-cost sensors, so that it is necessary to estimate it through the integration and processing of data acquired from low-cost devices or from the sensors installed on current vehicles (sensor fusion) [1,2]. In previous works, roll angle is estimated using different sensor types: inertial angle sensor and a gyroscope [1], angular rate and accelerometer sensors [2], lateral accelerometers and gyroscope [3–5], lateral and longitudinal accelerometers and yaw rate and roll rate sensors [4,6–8], on-board vehicle sensors and low-cost GPS [9,10] and lateral tire force sensors [11]. These observers are based on Kalman filter [5,6,9–11], robust estimators [2,3,7,8] or artificial intelligence techniques [6,12].

The design of RSC systems is a complex task as they have to fulfill some requirements, mutual to other safety vehicle systems:

- 1.-To acquire information from sensors which a high sampling frequency.
- 2.-To process sensor information in hard real time.
- 3.- To include actuators with fast-response time.
- 4.-To use low-cost systems in order to minimize the implementation cost in commercial vehicles.
- 5.- To develop an architecture which integrates all previous elements, guaranteeing high reliability and fault-tolerance.

The increase of computing power, the reduction of consumption and electric devices size, along with the high variety of communication technologies and networking protocols using Internet have yield to Internet of Things (IoT) development, being applied nowadays not only in smart manufacturing, healthcare, smart cities, but also, in transportation and smart vehicles [13–18]. Also, some research works have focused on hardware and software architectural problems related to this trend, and applied to the vehicular environment described before [19–22].

With the objective of design small and low-cost on-board systems for vehicle applications [19–22], it is necessary that they have enough accuracy and small processing time to increase the vehicle safety by the inclusion both estimators and controllers. These small computers will not only acquire the data but also must process this data to estimate the study variables. Raspberry Pi 3 Model B and Intel Edison are two popular small single-board computers, because they have flexibility, low price and high support from the internet community. There are studies that use these systems like a processing device [23], in [24] a fusion data for autonomous and transportation systems was performed through a Raspberry Pi. In [25] Raspberry Pi is used for detecting E.coli in real time. In [26] the dynamics of a human-powered vehicles was acquired through a Raspberry Pi. Finally, in [27] a study about inherent capabilities of the Raspberry Pi was carried out.

Like Raspberry Pi, there are many studies that use Intel Edison for the same purposes although it is not a single-board computer. In [28] integration with biomedical devices are used to acquired real-time vital parameters on neonates. In [29] a prototype to analyze geospatial data was created with Intel Edison. In [30] a system for smart home based on Intel Edison is proposed.

Previous studies demonstrate that the previous low-cost devices can provide data in real-time with the required precision and reliability [23].

On the other hand, Artificial Neural Networks (ANN), have been used to estimate vehicular characteristics in previous studies, like [31] where an ANN is used to estimate truck static weights by fusing weight-in-motion data, [32] where an ANN is used to estimate friction coefficient of wheel and rail in trains, [33] where an ANN is used to predict intersection crashes, or [34] where an ANN is used to estimate the traffic density and vehicle classification.

With the increase of computational power in small and embedded devices, ANNs have become computationally feasible to be used in such systems. This enhances the capabilities that IoT devices can provide [35–37]. However, in most cases it is necessary to assess that the fusion of data coming from low-cost devices and estimations provided by ANNs can fulfil the reliability and appropriateness requirements for using these technologies to improve overall safety in production vehicles.

The novelty reflected in this work is the analysis of the vehicle roll angle estimation using ANN and low-cost devices under high dynamic conditions, by following specific best practices and considerations. An architecture based on IoT has been developed integrating low-cost Inertial Motion Unit (IMU) and small single-board computer, that acquire the data from the IMU sensor and estimate the roll angle using ANNs. The outcome to the estimations have been compared with the measurements acquired by a high-end professional device (VBOX from Racelogic), used as the ground truth. Two different low-cost systems have been considered on this research systems (Raspberry Pi with IMU BNO0055 and Intel Edison with IMU LSM9DSO). These devices are compared in terms of estimation accuracy, processing time and reliability.

This article is structured as follows. In Section 2, the methodology is presented, including the experimental testbed design, experiments' definitions, and the data gathering and analysis. The experimental results and the calculation of the RMS error and processing time are presented in Section 3. Finally, in Section 4, the discussion and conclusion of the results and the method are exposed.

2. Methodology

This section begins with the description of the experimental testbed design defined to achieve the research goals; then, the experiments to gather the data required are specified; and finally, the data gathered to analyze real time estimators' performance and reliability are presented, indicating the data analysis strategies adopted in order to identify relevant results and conclusions.

2.1. Experimental Testbed Design

This research work experimental testbed design can be analyzed from two perspectives: hardware and software.

2.1.1 Hardware perspective

The experimental testbed is based on a Internet of Things (IoT) architecture, embedded in a vehicle. The complete architecture is packaged in a product that can be integrated in any vehicle, in this particular case, for testing purposes, a Mercedes-Benz van was used. The motivation to use this vehicle was to compare the results obtained during this research with those described in [6].

To properly perform the comparative analysis, three kits of sensors were considered:

- Reference, or ground truth kit. This kit is composed of a VBOX 3i GPS dual antenna data logger [38] connected to an IMU (Inertial Measurement Unit) from Racelogic. To accurately measure the roll angle, the two antennas form a 90-degree angle with respect to the traveling direction. For IoT kits' synchronization and data gathering, the VBOX controller is connected to a laptop embedded in the vehicle [6]. The installed sensors provide measurements for lateral acceleration, a_{ym} , longitudinal acceleration, a_x , yaw rate, $\dot{\psi}$, roll rate $\dot{\phi}$ and roll angle, ϕ . Given the nature of Racelogic VBOX devices, they need to be physically connected by wire to the experiments manager and among themselves.
- First low-cost sensor kit is composed by a Raspberry Pi 3 Model B [39,40] plus a low-cost Inertial Measurement Unit Shield [41].

- Second low-cost sensor kit is composed by an Intel Edison System-on-Chip [42] linked to a SparkFun “9 Degrees of Freedom” module [43].

The technical specifications of hardware elements considered for ground truth, Raspberry and Intel Edison kits are detailed in Table 1.

Table 1. Technical specifications of hardware elements included in the study.

	VBOX kit	Raspberry Pi kit	Intel Edison kit
RAM	2 GB ¹	1 GB	1 GB
CPU	Intel Core 2 Duo T8100 2.10GHz ¹	4xARM Cortex-A53, 1.2 GHz	4x Intel Atom Tangier x86 dual core processor +Intel Quark core
Power consumption	Max. 5.5 Watts ²	5 V @ < 1.5 W–6 W _o	3.3 V @ < 1 W
Dimensions	170 × 121 × 41 mm ²	85.60 × 56.5 mm	35.5 × 25 mm
Angular rate range	±150°/s	From ±125°/s–±2000°/s	From ±245°/s–±2000°/s
Acceleration range	±1.7 g	From ±2 g–±16 g	From ±2 g–±16 g
Angular rate resolution	0.01°/s	16 bits (From 0.003°/s for ±125°/s to 0.06°/s for ±2000°/s)	16 bits (From 0.007°/s for ±245°/s to 0.06°/s for ±2000°/s)
Acceleration resolution	0.01 g	14 bits (From 0.0002 g for ±2 g to 0.002 g for ±16 g)	14 bits (From 0.0002 g for ±2 g to 0.002 g for ±16 g)
Price	> 16.000 €	63,2 €	55,5 €

¹ This information corresponds to the laptop required to control the experiments and register log information in the ground truth kit.

² Only VBOX logger and IMU.

The IMU and the low-cost sensor kits were located in the vehicle’s center of mass, as is depicted in Figure 1. These three kits were also interconnected using a WiFi router which handles the communications among them, so the experiments can be synchronized, and the outcome of the tests can be gathered. According to [44], the accurate positioning of IMU and controller is essential for precision enhancement of low-cost kits.



Figure 1. Test vehicle (1) equipped with different low-cost systems (4, 5), VBOX data logger, 3 IMU sensors (2) and GPS dual-antenna (3).

2.1.2 Software perspective

A software architecture was designed to gather the data provided by the sensor kits in a synchronized way and to provide the datasets necessary to analyze the precision and performance of each kit considered for this research work. The main components of this architecture are shown in Figure 2.

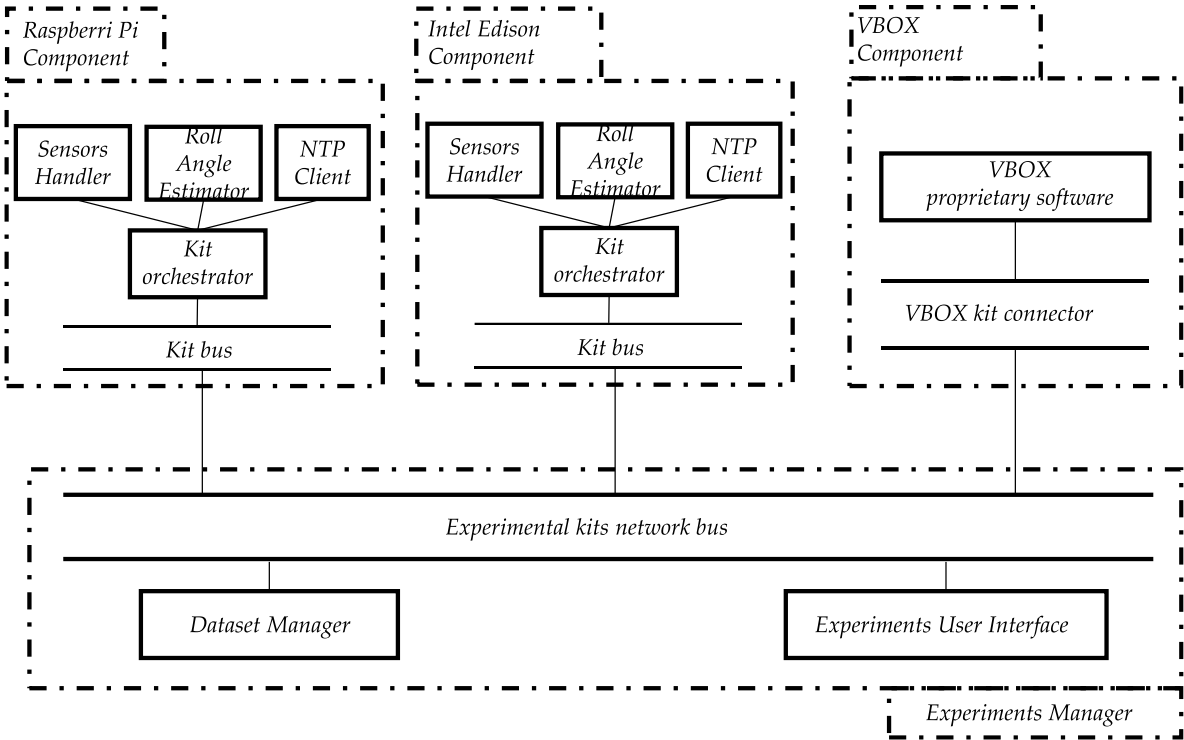


Figure 2. Testbed software design.

The *Experiments Manager* has the responsibility to provide a user interface to let the researcher start and stop the experiments and register the information coming from the experimental kits. It is developed in C++. The specific classes included in this component are:

- The *experimental kits network bus* is in charge of subscribing and unsubscribing the different experimental kits. Even more, it provides the possibility to send requests to the experimental kits (0, shutdown experimental kit; 1, keep running the experiment; 2, start the experiment; and

3, end the experiment) and to receive the information items provided by VBOX, Raspberry Pi and Intel Edison kits.

- *Dataset Manager* that is in charge of taking the data coming from the kits and storing them in a CSV file. The information stored includes gyroscope and accelerator data gathered with a sampling rate of 50 Hz.
- *Experiments User Interface* that provide the functionality to start and finish the experiments. It connects to the *experimental kits network bus* to start and finish an experiment in a synchronized way for all the experimental kits connected.

The *VBOX Component* is in charge of gathering the information provided from the Racelogic IMU sensor and GPS dual antenna data. It is developed in C#. The specific classes included in this component are:

- The *VBOX kit connector* oversees publishing the experimental kit, receiving the orders from the experimental kits orchestrator connected and send to it information obtained during the experiment.
- The *VBOX proprietary software* is in charge of managing the information received during the experiment execution.

The software to manage the Intel Edison experimental kit is in charge of gathering the information provided by gyroscope and accelerometer included in its hardware architecture. This component is implemented in C++. The specific classes included in this component are:

- The *kit bus* that is in charge of publishing the experimental kits in the network, receiving the requests form *experimental kits network bus* and send to the *kit orchestrator* the orders for starting and stopping the experiment.
- The *kit orchestrator* is in charge of creating an empty data structure to store the results in RAM memory after receiving the “start experiment” signal from the *kit bus*, and sending the data structure having the data gathered during the experiment to the *Experiments Manager* for storage purposes, after receiving the “end experiment” signal. The information sent is routed through the *kit bus* and the *experimental kits network bus* to reach the *Dataset Manager* in the *Experiments Manager*.
- The *Sensors Handler* is responsible of registering data items from sensors attending to the pre-configured sampling rate (50 Hz for this research work).
- The *Roll Angle Estimator* is a software component that implements an ANN to estimate the roll angle corresponding to the lateral acceleration, a_{ym} , the longitudinal acceleration, a_x , the yaw rate, ψ , and the roll rate $\dot{\phi}$ as input variables. A more detailed description of this estimator is provided in 2.1.3.
- The *NTP Client* is in charge of registering the actual date-time in the hardware controller of the experimental kit to ensure that all the kits in the testbed have the same date-time. This enables and eases comparison of results during the data analysis stage in this research work.

The Raspberry Pi kit has the same class structure as the Intel Edison kit. Sensors drivers were developed in C++ due to the recommendations provided in [23], and trying to keep the code as much similar as possible to maximize objectivity in the comparison of performance results against other devices with different hardware architectures (as the Intel Edison).

Raspberry Pi 3 and Intel Edison kits present wireless communication interfaces that ease the connectivity among components and allow to place them in virtually any point of the vehicle without worrying about setting up specific communication wires. Even more, the sensors used by these low-cost platforms are straightforwardly attached to the development boards by using the GPIO ports. By means of a wireless (IEEE 802.11g) access point, they can be connected to the experiments manager, which signals them their operation mode via a TCP socket connection.

2.1.3 Vehicle Roll Angle Estimator using Neural Networks

This software component uses ANNs to estimate the vehicle roll angle. The proposed model employs a Back-Propagation (BP) algorithm, which is one of the most widely-used methods for training a neural network. The specific neural network considered has been implemented as described in [6]. The architecture of the BP neural network is shown in Figure 3. The ANN has a single hidden layer of 15 neurons, four inputs (the lateral acceleration, a_{ym} , the longitudinal acceleration, a_x , the yaw rate, $\dot{\psi}$ and the roll rate $\dot{\phi}$) and one output (the vehicle roll angle, ϕ).

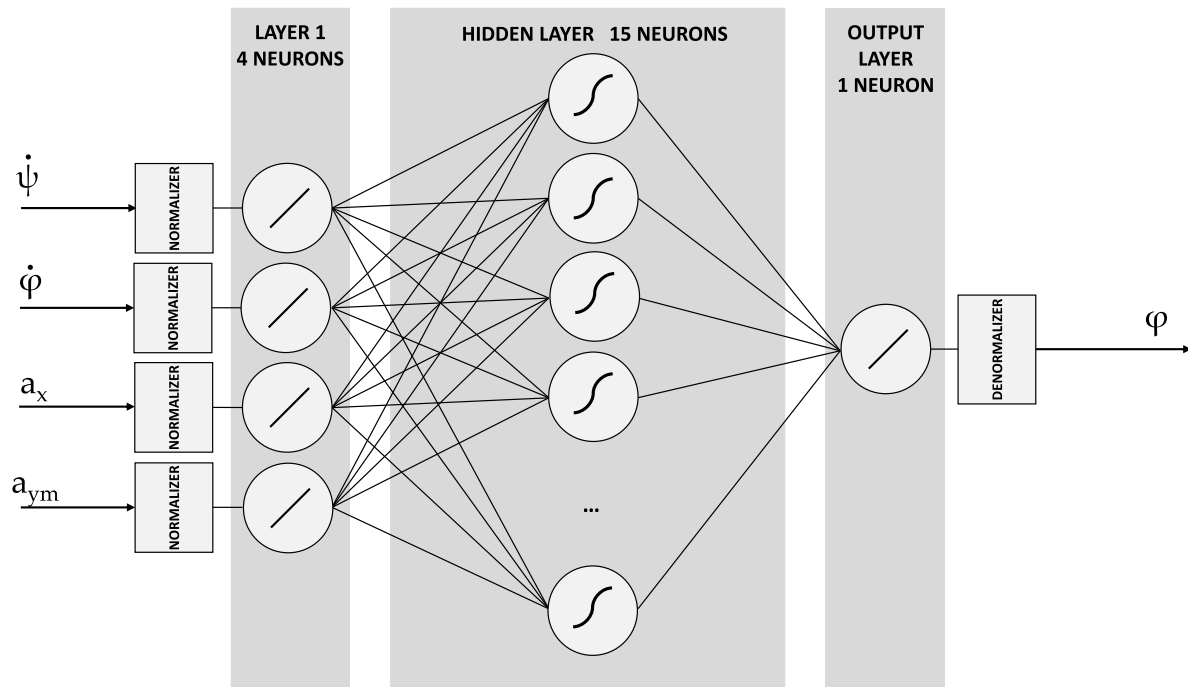


Figure 3. Artificial neural network architecture.

The contribution provided by this research work to the Neural Network considered in [6] is its implementation in C++ using the FANN framework [45] to obtain an estimator component able to satisfy the real-time restrictions related to embed this estimator in a control unit installed in an operating vehicle.

2.2. Experiments Definition

The hypotheses to evaluate during this research work are the following:

- H1: The roll angle estimated (ϕ_e) by the low-cost sensor kits is similar to the roll angle provided (ϕ_a) by expensive experimental kits (i.e., VBOX-based kits).
- H2: The performance of the low-cost sensor kits (i.e., Raspberry Pi 3 Model B and Edison Kits) estimating the roll angle achieves the levels required for real-time processing embedded in operating vehicles.

To evaluate the previous hypothesis, five controlled experiments were executed several times (see Table 2). The experiments consider typical maneuvers such as J-turn and lane change maneuvers. These maneuvers are the most common used to test the vehicle's behavior. Furthermore, a long test simulating a general execution has been done. In the experiments carried out, the lateral acceleration and the roll rate are the variables that have suffered higher variation, and for this reason, these variables are considered in order to analyze the accuracy of different devices.

The experimental tests have been carried out in Leganes (Madrid, Spain) using a Mercedes Sprinter, as can be seen in Figure 4 during a period where the setting had no traffic restrictions interfering with the appropriate execution of the considered experiments.

Table 2. Experiments proposed.

Id	Description	Times	Purpose	Variables to Observe
1	Vehicle simulates a normal circulation behavior, between 20 and 50 Km/h. Several curves were taken, and the vehicle was at the most appropriate speed for the road and conditions	1	(1) Estimator Accuracy (2) Estimator Performance	$a_{ym}, a_x, \dot{\psi}, \dot{\phi}, \varphi_a, \varphi_e$
2	Vehicle takes a roundabout with a radius of around 20 m at a constant speed inferior to 40 Km/h.	3	(1) Estimator Accuracy (2) Estimator Performance	$a_{ym}, a_x, \dot{\psi}, \dot{\phi}, \varphi_a, \varphi_e$
3	Vehicle takes a roundabout with a radius of around 20 m at a constant speed higher than 40 Km/h.	3	(1) Estimator Accuracy (2) Estimator Performance	$a_{ym}, a_x, \dot{\psi}, \dot{\phi}, \varphi_a, \varphi_e$
4	Vehicle performs a lane change a constant speed inferior to 40 Km/h.	3	(1) Estimator Accuracy (2) Estimator Performance	$a_{ym}, a_x, \dot{\psi}, \dot{\phi}, \varphi_a, \varphi_e$
5	Vehicle performs a lane change a constant speed higher than 40 Km/h.	3	(1) Estimator Accuracy (2) Estimator Performance	$a_{ym}, a_x, \dot{\psi}, \dot{\phi}, \varphi_a, \varphi_e$



Figure 4. Experiments' context (Map scale 1:7800 cm).

2.3. Data Gathering and Analysis

The data obtained for each of the previously-defined experiments (see Table 2) were stored by the controller of each kit in a CSV formatted file, identifying the experiment and its execution date and time. The variables considered were the lateral acceleration, a_{ym} , the longitudinal acceleration, a_x , the yaw rate, $\dot{\psi}$ and the roll rate $\dot{\phi}$, the actual roll angle φ_a (only obtained in VBOX kit, that acts

as the ground truth, together with the GPS coordinates) and the estimated roll angle, φ_e (only calculated by low-cost sensor kits).

The measures were gathered according to the sampling rate stated for the experiments, which was 50 Hz. Figure 5 and 6 present an example of the data gathered in each experiment.

time	lat	long	velocity	Yaw_Rate	X_Accel	Y_Accel	Roll_Rate	Roll_Angle
144505,53	2421,47226	224,51779	16,242	3,30E+01	-1,36E-01	-2,57E-03	-1,14E+00	-1,87E-01
144505,54	2421,47228	224,51779	17,02	1,59E+01	-1,36E-01	-5,92E-03	-1,05E+00	-1,07E-01
144505,55	2421,47231	224,51779	17,15	1,03E+01	-1,38E-01	-1,33E-03	-1,09E+00	-1,23E-01
144505,56	2421,47233	224,5178	17,094	-1,60E+01	-1,35E-01	-2,70E-03	-8,80E-01	-3,57E-01
144505,57	2421,47236	224,5178	16,742	4,00E-01	-1,35E-01	-3,14E-03	-7,90E-01	-3,97E-01
144505,58	2421,47238	224,51781	16,501	-3,64E+01	-1,38E-01	1,94E-03	-1,07E+00	-3,41E-01
144505,59	2421,47241	224,51781	16,649	9,40E+00	-1,39E-01	8,57E-04	-1,26E+00	-1,35E-02
144505,6	2421,47243	224,51782	16,594	5,80E+00	-1,36E-01	9,13E-04	-1,05E+00	-9,54E-03
144505,61	2421,47246	224,51782	16,724	3,90E+00	-1,30E-01	3,58E-03	-4,40E-01	-2,89E-01
144505,62	2421,47248	224,51782	16,687	2,20E+00	-1,30E-01	2,90E-03	-4,90E-01	-2,35E-02
144505,63	2421,47251	224,51783	16,705	2,50E+00	-1,36E-01	-3,71E-03	-4,30E-01	-3,07E-01
144505,64	2421,47253	224,51783	16,612	3,69E+01	-1,29E-01	-1,01E-02	-1,20E-01	-4,96E-01
144505,65	2421,47255	224,51783	16,261	-2,27E+01	-1,33E-01	-6,70E-03	-1,30E-01	-3,89E-01
144505,66	2421,47258	224,51784	16,001	3,23E+01	-1,35E-01	-2,07E-03	-1,60E-01	-2,17E-01
144505,67	2421,4726	224,51784	16,316	-4,17E+01	-1,42E-01	-2,49E-03	-3,00E-02	-3,28E-01
144505,68	2421,47263	224,51785	16,427	-1,89E+01	-1,33E-01	-1,88E-04	6,00E-02	-3,25E-01
144505,69	2421,47265	224,51785	16,39	7,40E+00	-1,35E-01	3,36E-03	-4,00E-02	-3,78E-01

Figure 5. Example of CSV file contents for data registered by VBOX kit during the experiments' execution.

TIME	YAW_RATE	ROLL_RATE	ACCEL_X	ACCEL_Y	ANN_ESTIMATED_ROLL_ANGLE
2018-03-13 19:03:32:776	0,0358	-0,13927	-0,0289757	0,014739	1,7519886494
2018-03-13 19:03:32:796	-0,0181	-0,27011	-0,0563395	0,025115	1,7087090015
2018-03-13 19:03:32:816	0,0447	-0,15048	-0,0588586	0,00833	1,6761229038
2018-03-13 19:03:32:836	0,0627	-0,03085	-0,0623155	0,035674	1,7133924961
2018-03-13 19:03:32:857	0,1165	-0,14301	-0,0955382	-0,011506	1,7465906143
2018-03-13 19:03:32:877	-0,0988	-0,19908	-0,0723936	0,001738	1,7288713455
2018-03-13 19:03:32:897	-0,3949	-0,18413	-0,0798355	0,018523	1,7141706944
2018-03-13 19:03:32:917	-0,0809	-0,21404	-0,0555187	0,013945	1,7598326206
2018-03-13 19:03:32:938	-0,3052	-0,14674	-0,0326083	0,027495	1,7169997692
2018-03-13 19:03:32:958	0,0268	0,06261	-0,0221789	0,018645	1,6781463623
2018-03-13 19:03:32:978	-0,2514	0,05513	-0,0274522	0,015593	1,7460863590
2018-03-13 19:03:32:998	-0,8345	0,18224	-0,0227645	0,013274	1,7259330750
2018-03-13 19:03:33:18	-0,7897	0,27196	-0,0401674	0,021758	1,7189614773
2018-03-13 19:03:33:39	-0,8345	0,36168	-0,1013395	0,03543	1,7645335197
2018-03-13 19:03:33:59	-0,6730	0,29439	-0,063312	-0,006868	1,7884426117
2018-03-13 19:03:33:79	-0,6461	0,04392	-0,0883901	0,009124	1,7162549496
2018-03-13 19:03:33:99	-0,6910	0,17850	-0,0760262	0,017974	1,6603155136

Figure 6. Example of CSV file contents for data registered by low cost sensor kits during the experiments' execution.

The accuracy of roll angle estimation using NN and data collected by low-cost sensor kits was calculated comparing these data against the roll angle obtained from GPS-dual antenna by the Racelogic VBOX (Ground Truth). In Section 3, the results obtained in the different tests will be presented.

2.4. Threats to Validity

To analyze the validity of the results obtained in this experimental work, several threats were considered prior to the experiments' execution:

- Internal validity is the extent to which a causal conclusion based on the experiments defined is appropriate avoiding the introduction of systematic errors in the data used to determine the results and conclusions.
- External validity is the extent to which the results of a study can be generalized to other situations.

(A) Internal validity: In this research, the only factor contributing to internal validity is related to the specific sensors used to configure each low-cost kit and the software components implemented to manage the required data:

- The first threat was mitigated using two different kits for each type, in order to prevent possible errors produced by sensors that may provide incorrect values. Even more, all the kits considered were tested in a static environment, preconfiguring the corresponding calibrations to assure that specific sensors included were providing appropriate data.
- Regarding the software components, the possible threats were mitigated implementing an exhaustive unit testing process. It ensures that each functionality is properly processing the received values and that synchronization among devices is correctly implemented.
- The ANN based estimators were properly trained and compared with information coming from experiments carried out in previous research works [6,23].
- To verify the validity of the results, each maneuver was repeated, at least, three times consecutively.

(B) External validity: In the scope of this research, the factors that influence the external validity are related to the replication of this experiment. These replications must consider several relevant factors: sensors and controllers included in the experimental kits, their location in the vehicle and road conditions:

- Regarding sensors and controllers, the possible threats were mitigated using sensors and controllers available on the market having regular features [46–48]. On this regard, the conclusions obtained are valid for the low-cost sensors currently available on the market, and as the technology is always improving the prior conditions, the conclusions can be applied in forthcoming low-cost sensors.
- Regarding vehicle conditions, the threats were related to the appropriate location of experimental kits in order to ensure equal conditions among them. This threat was mitigated creating a box to put the three sensor kits in a 3D printed enclosure (See Figure 1, box 4) that was located in the vehicle gravity center. These issues are important to replicate this experiment properly in other vehicles. In this research, as the experiments have been carried out with a real vehicle, it has been difficult to maintain the same exact driving conditions related to steering wheel angle and velocity. In spite of this problem, similar tests for typical maneuvers have been carried out.
- Regarding road conditions and experiments execution, the threats were related to the representativeness of the scenarios considered. This threat was mitigated considering a road without relevant slope variations and including different experiments with different types of directions, constant and variable speed.

3. Results

As is indicated in Section 2, a Mercedes Sprinter was used for this work (see Figure 1). Three different kind of experiments were carried out: two different maneuvers, J-Turn and lane change, and a normal circulation test. For J-Turn and lane change maneuvers, five similar tests were performed in order to assess the validity of the results.

3.1 Test 1. J-turn

The first test is performed in a roundabout with a radius of 22 m (see Figure 7) at a constant speed (close to 40 Km/h). Figure 8 shows the roll angle estimated by the Raspberry Pi (blue) and Intel Edison (green). In order to verify the estimation accuracy, results have been compared with the roll angle measured with the VBOX GPS dual antenna (yellow), which is considered as the ground truth. Estimations are very similar in both devices, and that the usage of low-cost devices do not impact in the ANN estimator performance.

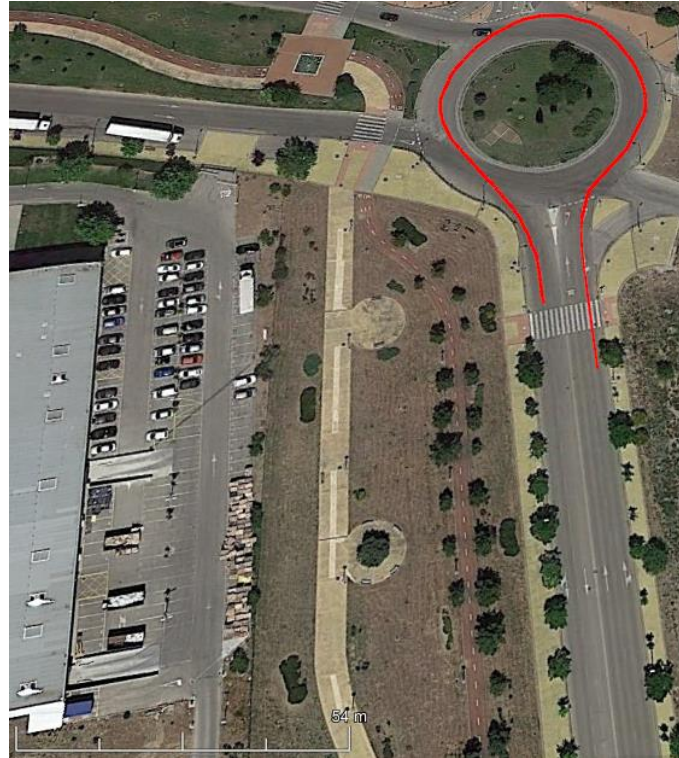


Figure 7. Test 1: Map and vehicle trajectory (Map scale 1:2100 cm)

To quantify this impact, the norm, the root mean square (RMS) and maximum errors have been calculated. The norm error as a function of time is calculated as follows [7]:

$$E_t = \frac{\varepsilon_t}{\sigma_t} \cdot 100 \quad (1)$$

Where

$$\varepsilon_t^2 = \int_0^T (\theta_{GT} - \theta_{lc})^2 dt \quad (2)$$

$$\sigma_t^2 = \int_0^T (\theta_{GT} - \mu_{GT})^2 dt$$

θ_{GT} represents the ground truth data, θ_{lc} represents the low-cost sensor data and μ_{GT} is the mean value of the ground truth data obtained during the period T.

Table 3 contains the errors measured. To verify the validity of the results, three similar tests for the J-turn maneuver have been carried out. To quantify the dispersion of data values, the standard

deviation has been included for the RMS error (see Table 3). The results show that the error is very similar in both devices and it is higher than the estimated roll angle using VBOX IMU data. The difference between Raspberry Pi 3 Model B and Intel Edison for the norm and RMS error is about 3% and 0.05°, respectively. Concerning the maximum errors, the difference is about 0.3°.

Table 3. Test 1. Errors of estimated roll angle on Raspberry Pi and Intel Edison compared with the measured roll from VBOX (ground truth).

	Roll angle		
	Norm Error	RMS error	Maximum Error
	(%)	(°)	(°)
Raspberry Pi 3 Model B	62.09	0.7405±0.0823	3.54
Intel Edison	65.74	0.7965±0.0743	3.84
Racelogic VBOX IMU	52.22	0.5792±0.0322	2.74

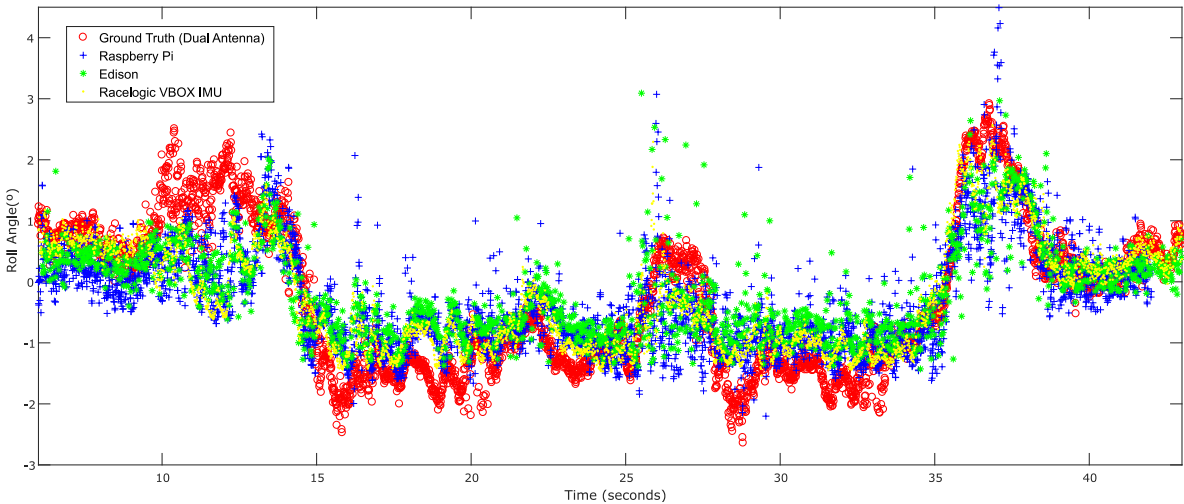


Figure 8. ANN estimated + ground truth roll angle for J-Turn maneuver

An important aspect to consider in this kind of system is the temporal performance and real time constraints. For the given case, the system needs to be able of processing the inputs and apply the ANN estimator in less than 20 ms, corresponding to the sampling rate of 50 Hz forced by the low-cost sensors.

Figures 9 and 10 show the relationship between the sensors’ measured data processing time (normalization + ANN estimation + denormalization) for both Intel Edison and Raspberry Pi 3 Model B respectively, and the established threshold corresponding to the sampling rate (50 Hz).

In table 4, a comparison of time performance between Intel Edison and Raspberry Pi 3 Model B is presented. To quantify the performance of the devices, the mean and maximum processing times have been calculated. It has been also calculated the mean deviation to assess the stability of the devices. Results show that both devices estimate roll angle four orders of magnitude lower than the required sample rate threshold of 20 ms. Results show that the processing times for Raspberry Pi 3 Model B are higher than the Intel Edison ones. Concerning the mean and maximum times, the differences are about 0.5e-3 ms and 12 ms respectively. Regarding Mean Deviation, the difference is about 0.008 ms, so it is possible to conclude that results are homogeneous as far as performance and response times are concerned.

Table 4. Test 1. Processing time on Raspberry Pi and Intel Edison.

	Processing Time		
	Maximum	Mean	Mean deviation
	(ms)	(ms)	(ms)
Raspberry Pi Model 3	13.09	18.06e-3	13.1e-3
Intel Edison	1.19	13.87e-3	5.1e-3

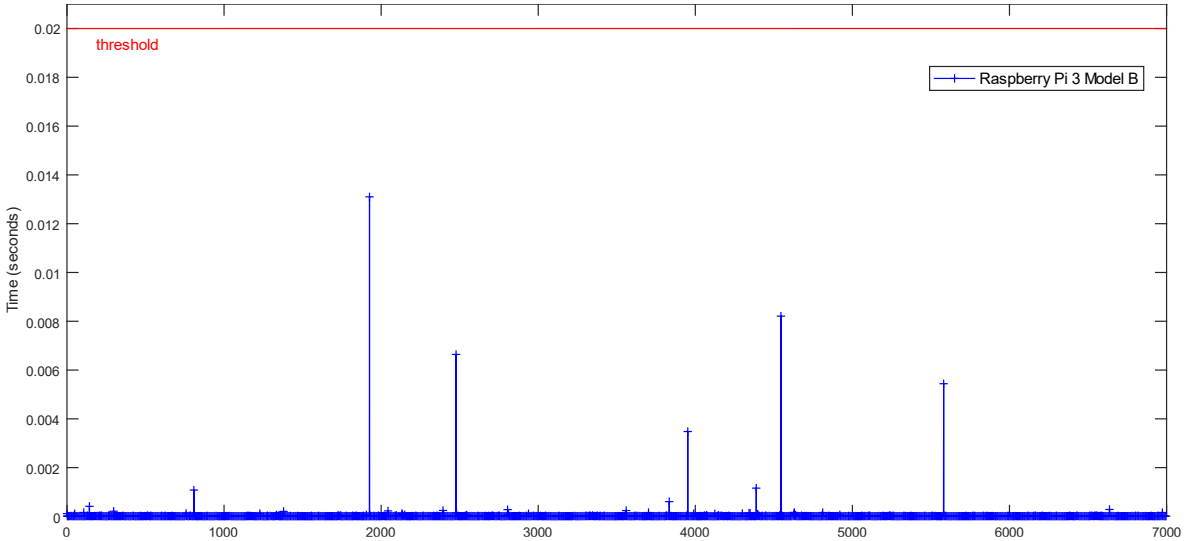


Figure 9. Test 1. Processing time of each iteration for Raspberry Pi

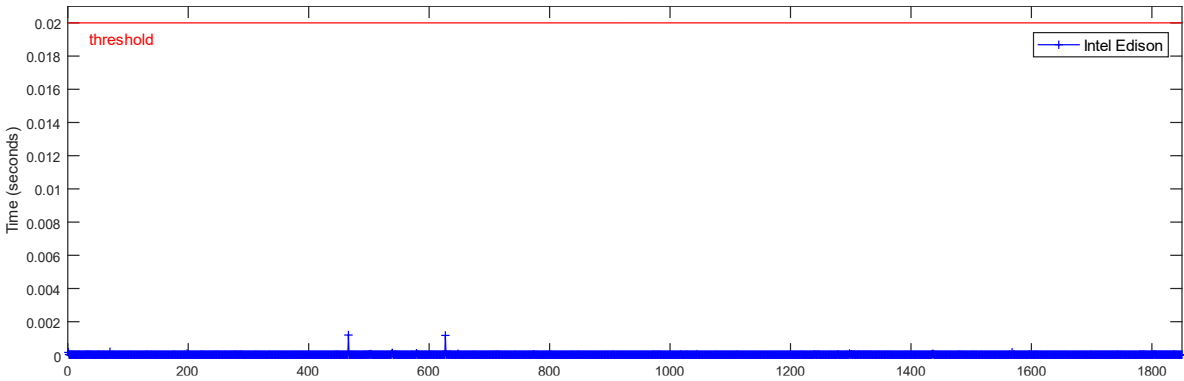


Figure 10. Test 1. Processing time of each iteration for Intel Edison

3.2 Test 2. Double Lane Change

The second test is carried out in a straight line when the vehicle does a slalom at constant speed (See Figure 11). Figure 12 shows the roll angle estimated by the Raspberry Pi (blue), Intel Edison (green) and the data provided by the Racelogic IMU (yellow). In order to verify the accuracy of the estimation, they have been compared with the roll angle measured with the GPS dual antenna of VBOX, which is considered as the ground truth. It can be seen that the estimation is very similar in both cases and that using low-cost devices do not impact in the ANN performance.

To quantify this impact, the norm, RMS and maximum errors have been calculated (see Table 5). To verify the validity of the results, three similar tests for the Lane Change maneuver have been carried out. To quantify the dispersion of data values, the standard deviation has been included for the RMS error. The results show that the errors are very similar in both devices and they are higher than the estimated roll angle using VBOX IMU data. The difference between Raspberry Pi 3 Model B and Intel Edison for the norm and RMS errors are about 0.6% and 0.03°, respectively. Concerning the maximum errors, the difference is about 0.2°.

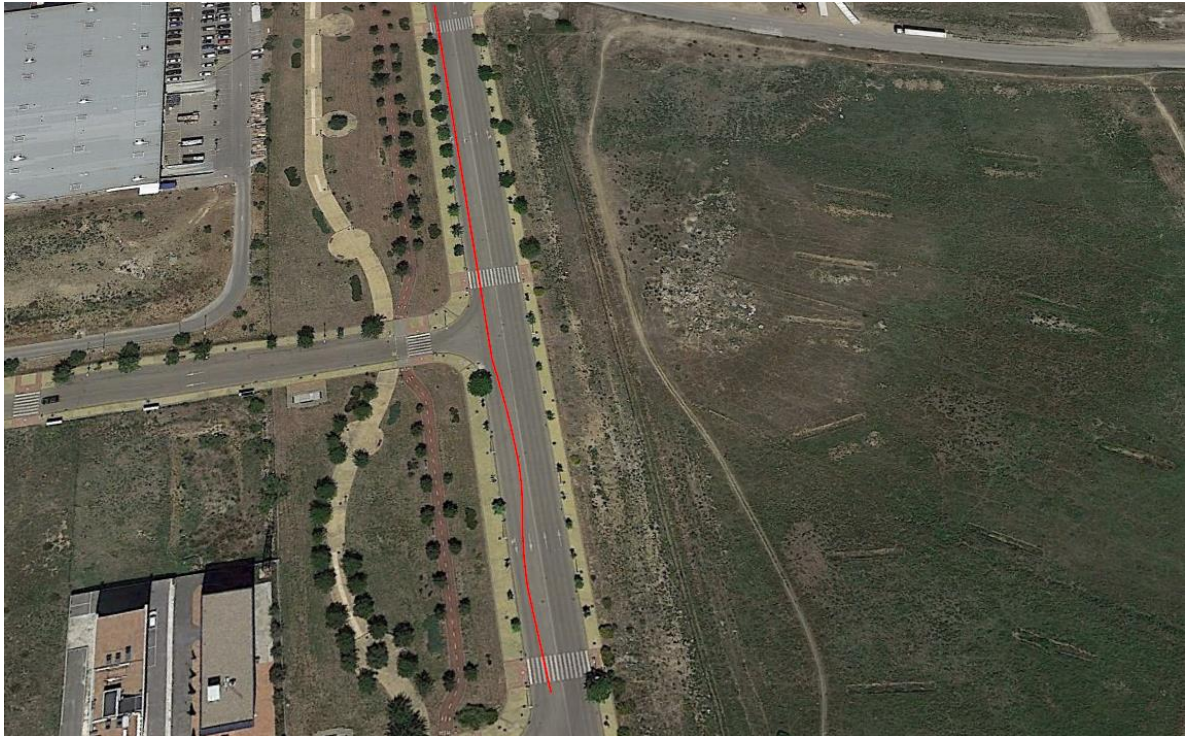


Figure 11. Test 2: Map and vehicle trajectory (Map scale 1:2100 cm)

Table 5. Test 2. Errors of estimated roll angle on Raspberry Pi and Intel Edison compared with the measured roll from VBOX (ground truth).

	Roll angle		
	Norm Error	RMS error	Maximum Error
	(%)	(⁰)	(⁰)
Raspberry Pi Model 3	85.37	0.5302±0.0681	2.54
Intel Edison	85.98	0.5075±0.0432	2.36
Racelogic VBOX IMU	72.84	0.4521±0.0215	1.95

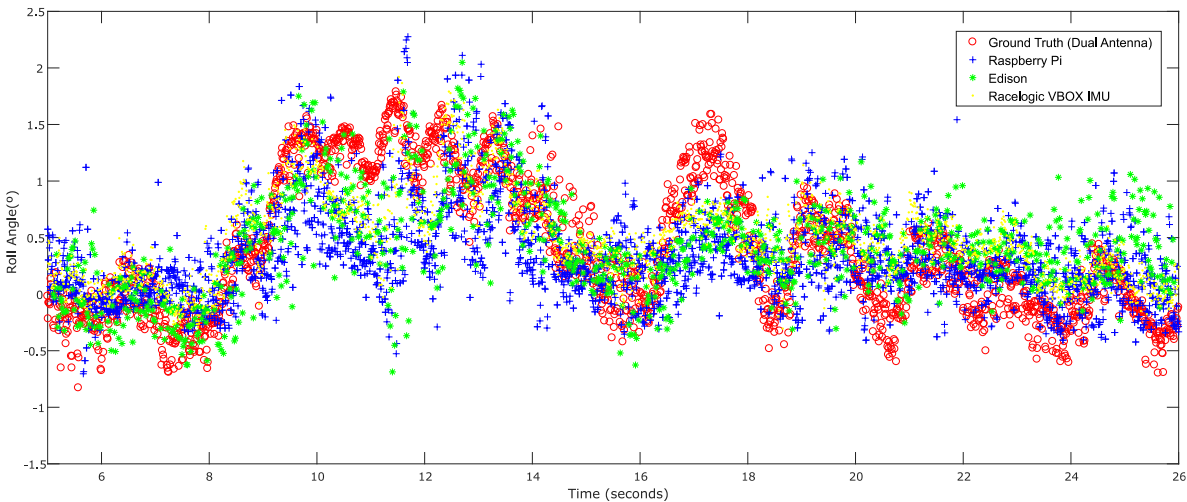


Figure 12. ANN estimated + ground truth roll angle for double lane change

Figures 13 and 14 show the relationship between the sensors’ measured data processing time (normalization + ANN estimation + denormalization) for both Intel Edison and Raspberry Pi 3 Model B, respectively, and the established threshold corresponding to the sampling rate (50 Hz).

In Table 6, a time performance comparison between Intel Edison and Raspberry Pi 3 Model B is presented. To quantify this performance for both devices, the mean and maximum processing times have been calculated. It has been also calculated the mean deviation in order to assess the stability of the devices. Results show that both devices estimate roll angle four orders of magnitude lower than the required sample rate threshold of 20 ms. Results show that the processing times for Raspberry Pi 3 Model B are higher than the Intel Edison ones. Concerning to mean and maximum times the difference is about $0.7e^{-3}$ ms and 7.9 ms. Regarding mean deviation, the difference is about $3.8e^{-3}$ ms, so it is possible to conclude that results are homogeneous as far as performance and response times are concerned.

Table 6. Test 2. Processing time on Raspberry Pi and Intel Edison.

	Processing Time		
	Maximum	Mean	Mean deviation
	(s)	(s)	(s)
Raspberry Pi Model 3	$8.02e^{-3}$	$12.32e^{-6}$	$6.1e^{-6}$
Intel Edison	$0.13e^{-3}$	$11.59e^{-6}$	$2.3e^{-6}$

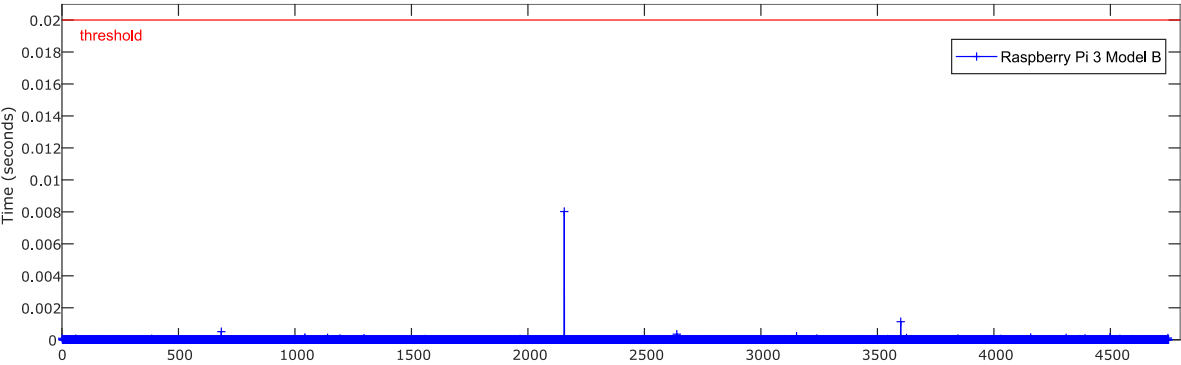


Figure 13. Test 2. Processing time of each iteration for Raspberry Pi

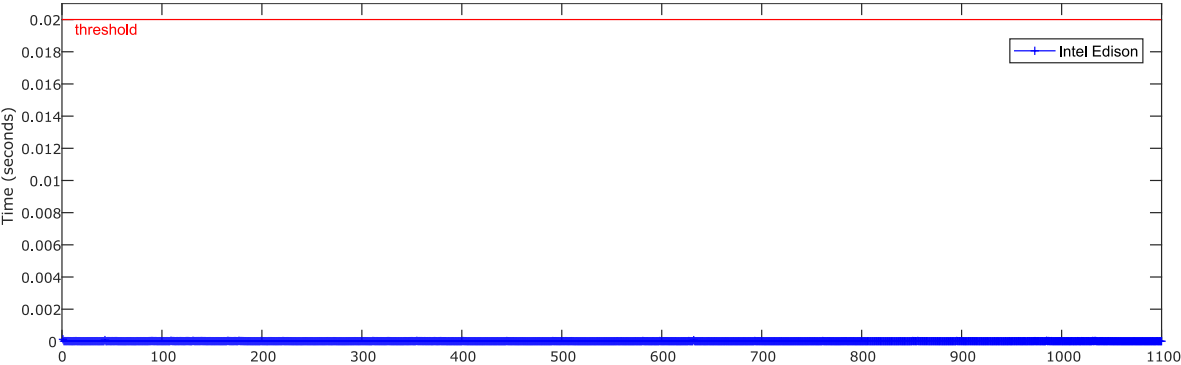


Figure 14. Test 2. Processing time of each iteration for Intel Edison

3.3. Test 3. General Circulation

This last test is carried out in the circuit shown in Figure 4. In this test not only J-Turn and Lane Change maneuvers are performed, but also the course of a real circuit under usual circulation conditions. The vehicle is driven at every moment with the most appropriate speed for the road and the situation, doing severe maneuvers at low and medium speed circulation (between 20 and 60 Km/h), and smooth movements.

Figure 15 shows the roll angle estimated by the Raspberry Pi (blue), Intel Edison (green) and the data provided by the Racelogic IMU (yellow) that is considered as ground truth. In this test the calculated error is higher than the other two tests. This kind of tests are prone to suffer noise, and as is indicated in [23], the low-cost sensors used are very sensitive to noise. Table 7 shows that the error is higher in Intel Edison than in Raspberry Pi, in this case Intel Edison present some atypical data. The difference of the norm and RMS error is about 96% and 0.9° respectively. Concerning the maximum errors, the difference is about 3° .

Table 7. Test 3. Errors of estimated roll angle on Raspberry Pi and Intel Edison compared with the measured roll from VBox (ground truth).

	Roll angle		
	Norm Error	RMS error	Maximum Error
	(%)	($^{\circ}$)	($^{\circ}$)
Raspberry Pi Model 3	107.91	1.0321	5.92
Intel Edison	135.87	1.3297	4.41
Racelogic VBox IMU	92.09	0.9431	5.29

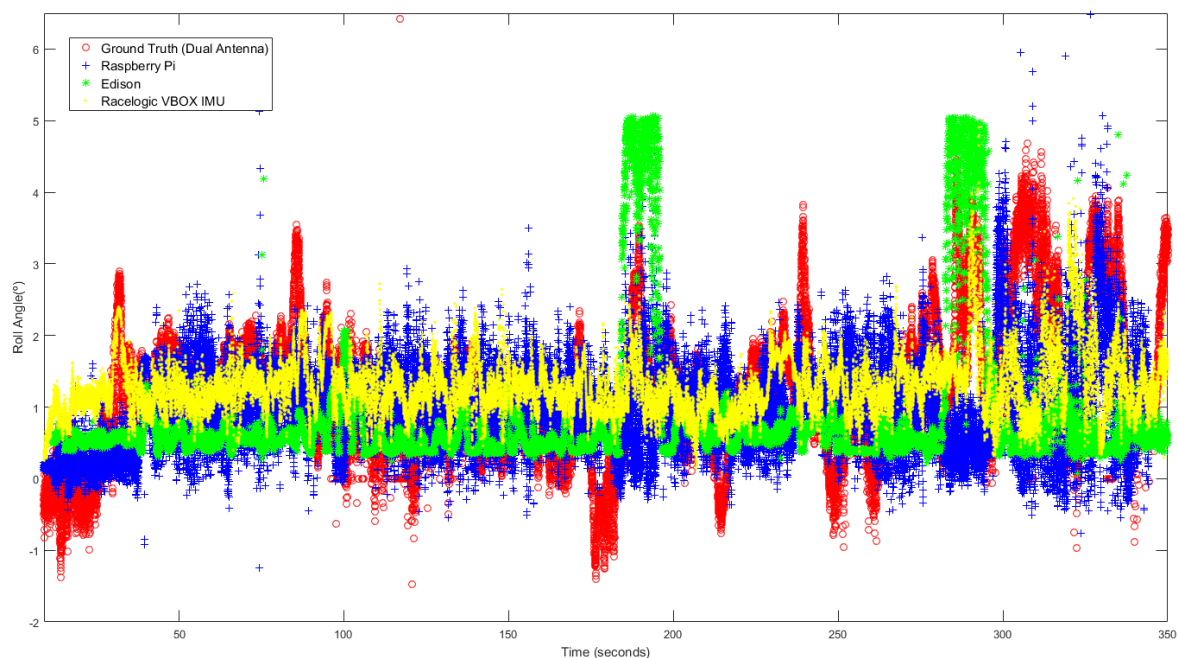


Figure 15. ANN estimated + ground truth roll angle for general circulation

Figures 16 and 17 show the relationship between the sensors measured data processing time (normalization + ANN estimation + denormalization) for both Intel Edison and Raspberry Pi 3 Model B respectively, and the established threshold corresponding to the sampling rate (50 Hz).

In table 8, a comparison of time performance between Intel Edison and Raspberry Pi 3 Model B is presented. To quantify the performance of the devices, the mean and maximum processing time have been calculated. It has been also calculated the mean deviation in order to assess the stability of the devices. As in previous tests, it is observed that both devices estimate the roll angle four

orders of magnitude lower than the required sample rate. Results show that the processing times for Raspberry Pi 3 Model B are higher than the Intel Edison ones. Concerning the mean and maximum times, the difference is about 0.7×10^{-3} ms and 5.1 ms, respectively. Regarding Mean Deviation, the difference is about 0.1×10^{-6} s, so it is possible to conclude that results are homogeneous as far as performance and response times is concerned.

Table 8. Test 3. Processing time on Raspberry Pi and Intel Edison.

	Processing Time		
	Maximum	Mean	Mean deviation
	(ms)	(ms)	(ms)
Raspberry Pi Model 3	14.88	14.58×10^{-3}	5.8×10^{-3}
Intel Edison	9.54	15.09×10^{-3}	5.9×10^{-3}

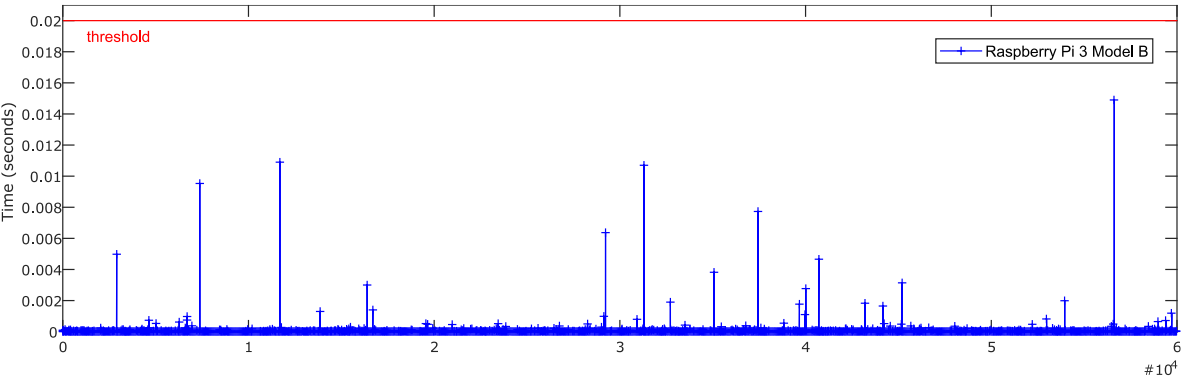


Figure 16. Test 3. Processing time of each iteration for Raspberry Pi

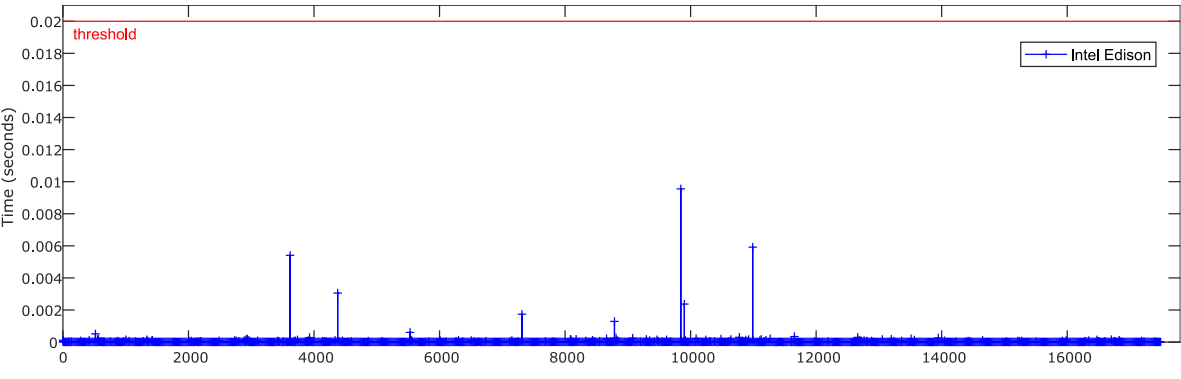


Figure 17. Test 3. Processing time of each iteration for Intel Edison

4. Discussion

The following discussion is focused on the precision and the performance of the low-cost devices.

4.1. Precision

Results show the estimation of the roll angle obtained from low-cost devices is like the measurements directly obtained from Racelogic VBox. Figure 15 shows that there exists noise in the estimated values. One reason is that the low-cost sensors are more prone to noise as is indicated

in [23] and ANNs are very sensitive to noise. As future work, it is planned to integrate filters via software components to solve the noise related issue.

Despite the noise influence, the average RMS error in Intel Edison and in Raspberry Pi is 0.8°.

4.2. Processing capability

The temporal performance and real time constraints are main aspects to consider in order to integrate estimators and controllers in embedded low-cost devices. The results show that the processing time to get the data, execute its normalization, perform the roll angle estimation via ANN and the denormalization of the outcome, is four orders of magnitude lower than the required sample rate threshold of 20 ms. The average mean processing time is 14.5×10^{-3} ms for Raspberry Pi 3 Model B and 13×10^{-3} ms for Intel Edison. This difference allows to integrate filters in order to reduce the noise in data collected from the sensors as it is indicated in 4.1 and to develop and embed more complex estimators and controllers.

To minimize the processing time and allow an objective performance comparison among the low-cost kits, some optimizations were considered before deploying and executing the software components. The most relevant optimizations include:

- Development of same C++ source code for Intel Edison and Raspberry Pi 3 Model B. Usage of same compiler (gcc version 6.3.0) in both platforms, and the same linker and compiler flags, considering the maximum optimization level for speed (-O3 [49]). These additional optimizations perform, among others, predictive commoning optimization, this is, reusing computations (especially memory loads and stores) performed in previous iterations of loops, with result beneficial considering the processor caches in both Intel Edison and Raspberry Pi 3 Model B.
- Usage of light and optimized Fast Artificial Neural Network Library (FANN [45]), version 2, compiled directly in the platforms using *cmake*, and the last source code revision from GitHub [50], that present among its multiple benefits the cache optimization for extra speed.
- Multiple revisions of source code to keep it clean and simple. Algorithmic optimizations to keep a low-profile memory usage, and increased performance (i.e. avoiding copies of objects, like the ANN instance, by passing it by reference).

5. Conclusions

In accordance to the results obtained, it can be concluded that low-cost experimental kits including embedded ANN estimators provide roll angle estimations very close to actual values. Even more, Intel Edison and Raspberry Pi 3 Model B have enough computing capabilities to successfully run roll angle estimation based on ANNs to determine rollover risks situation fulfilling real-time operation restrictions stated for this problem.

The results can be used to design, implement and test an efficient, versatile and scalable low-cost hardware/software architectures able to be integrated in commercial vehicles.

Even more, the performance levels achieved indicate the possibility to embed, in the low-cost experimental kits, more complex estimators using a sensor fusion approach to obtain roll angle estimations closer to the actual values based on Kalman filters, combining neural networks and Kalman filters and considering deep learning techniques including other parameters obtained from other sources (i.e. road characteristics). In this line, more complex estimators for other variables (such side slip, etc.) could be integrated in low-cost experimental kits to improve the vehicle stability control in real time.

Acknowledgments: This work might not have been possible had it not been for the funds provided by the Spanish Government through the projects TRA2013-48030-C2-1-R and TRA2008-05373/AUT.

Author Contributions: Javier García Guzmán, Lisardo Prieto González, Jonatan Pajares Redondo, Mat Max Montalvo Martínez and María Jesús L. Boada proposed the ideas. Lisardo Prieto González, Javier García Guzmán and Mat Max Montalvo Martínez designed the IOT architecture, communications and software components. Maria Jesús L. Boada, Mat Max Montalvo Martínez and Lisardo Prieto González implemented and tuned the ANN, Lisardo Prieto González and Jonatan Pajares Redondo developed the hardware setup. Javier García Guzmán, Lisardo Prieto González, Jonatan Pajares Redondo, Mat Max Montalvo Martínez and María Jesús L. Boada conceived and designed the tests. Jonatan Pajares Redondo, María Jesús L. Boada, Mat Max Montalvo Martínez and Lisardo Prieto González analyzed the data. Javier García Guzmán, Lisardo Prieto González, Jonatan Pajares Redondo and María Jesús L. Boada wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANN Artificial Neural Network

FANN Fast Artificial Neural Network (software library)

GPS Global Positioning System

IMU Inertial Measurement Unit

IoT Internet Of Things

RMS Root Mean Square

RSC Roll Stability Control (system)

References

1. Rajamani, R.; Piyabongkarn, D.; Tsourapas, V.; Lew, J. Real-time estimation of roll angle and CG height for active rollover prevention applications. In *American Control Conference, 2009. ACC'09.*; IEEE, 2009; pp. 433–438.
2. Tafner, R.; Reichhartinger, M.; Horn, M. Robust online roll dynamics identification of a vehicle using sliding mode concepts. *Control Eng. Pract.* **2014**, *29*, 235–246, doi:10.1016/j.conengprac.2014.03.002.
3. Zhao, L.; Liu, Z. Vehicle Velocity and Roll Angle Estimation with Road and Friction Adaptation for Four-Wheel Independent Drive Electric Vehicle Available online: <https://www.hindawi.com/journals/mpe/2014/801628/abs/> (accessed on Apr 16, 2018).
4. Suh, Y. S. Computationally Efficient Pitch and Roll Estimation Using a Unit Direction Vector. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 459–465, doi:10.1109/TIM.2017.2764358.
5. Zhang, S.; Yu, S.; Liu, C.; Yuan, X.; Liu, S. A Dual-Linear Kalman Filter for Real-Time Orientation Determination System Using Low-Cost MEMS Sensors. *Sensors* **2016**, *16*, 264, doi:10.3390/s16020264.
6. Vargas-Meléndez, L.; Boada, B. L.; Boada, M. J. L.; Gauchía, A.; Díaz, V. A Sensor Fusion Method Based on an Integrated Neural Network and Kalman Filter for Vehicle Roll Angle Estimation. *Sensors* **2016**, *16*, 1400, doi:10.3390/s16091400.
7. Boada, B. L.; Boada, M. J. L.; Vargas-Melendez, L.; Diaz, V. A robust observer based on H_∞ filtering with parameter uncertainties combined with Neural Networks for estimation of vehicle roll angle. *Mech. Syst. Signal Process.* **2018**, *99*, 611–623, doi:10.1016/j.ymssp.2017.06.044.
8. Boada, B. L.; Boada, M. J. L.; Diaz, V. A robust observer based on energy-to-peak filtering in combination with neural networks for parameter varying systems and its application to vehicle roll angle estimation. *Mechatronics* **2018**, *50*, 196–204, doi:10.1016/j.mechatronics.2018.02.008.

9. Bevely, D. M.; Ryu, J.; Gerdes, J. C. Integrating INS Sensors With GPS Measurements for Continuous Estimation of Vehicle Sideslip, Roll, and Tire Cornering Stiffness. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 483–493, doi:10.1109/TITS.2006.883110.
10. Jiang, G.; Liu, L.; Guo, C.; Chen, J.; Muhammad, F.; Miao, X. A novel fusion algorithm for estimation of the side-slip angle and the roll angle of a vehicle with optimized key parameters. *Proc. Inst. Mech. Eng. Part J. Automob. Eng.* **2017**, *231*, 161–174, doi:10.1177/0954407016644879.
11. Nam, K.; Oh, S.; Fujimoto, H.; Hori, Y. Estimation of Sideslip and Roll Angles of Electric Vehicles Using Lateral Tire Force Sensors Through RLS and Kalman Filter Approaches. *IEEE Trans. Ind. Electron.* **2013**, *60*, 988–1000, doi:10.1109/TIE.2012.2188874.
12. Boada, M. J. L.; Boada, B. L.; Gauchia Babe, A.; Calvo Ramos, J. A.; Vicente Diaz Lopez Active roll control using reinforcement learning for a single unit heavy vehicle. *Int. J. Heavy Veh. Syst.* **2009**, *16*, 412–430, doi:10.1504/IJHVS.2009.027413.
13. Wan, J.; Chen, B.; Imran, M.; Tao, F.; Li, D.; Liu, C.; Ahmad, S. Toward Dynamic Resources Management for IoT-Based Manufacturing. *IEEE Commun. Mag.* **2018**, *56*, 52–59, doi:10.1109/MCOM.2018.1700629.
14. Tsao, K.-C.; Lee, L.; Chu, T.-S.; Huang, Y.-H. A Two-Stage Reconstruction Processor for Human Detection in Compressive Sensing CMOS Radar. *Sensors* **2018**, *18*, 1106, doi:10.3390/s18041106.
15. Lau, B. P. L.; Wijerathne, N.; Ng, B. K. K.; Yuen, C. Sensor Fusion for Public Space Utilization Monitoring in a Smart City. *IEEE Internet Things J.* **2018**, *5*, 473–481, doi:10.1109/JIOT.2017.2748987.
16. Sohail, M.; Wang, L. 3VSR: Three Valued Secure Routing for Vehicular Ad Hoc Networks using Sensing Logic in Adversarial Environment. *Sensors* **2018**, *18*, 856, doi:10.3390/s18030856.
17. Jo, O.; Kim, Y. K.; Kim, J. Internet of Things for Smart Railway: Feasibility and Applications. *IEEE Internet Things J.* **2018**, *5*, 482–490, doi:10.1109/JIOT.2017.2749401.
18. Lam, A. Y. S.; Li, V. O. K. Opportunistic Routing for Vehicular Energy Network. *IEEE Internet Things J.* **2018**, *5*, 533–545, doi:10.1109/JIOT.2017.2752222.
19. Hermans, T.; Ramaekers, P.; Denil, J.; Meulenaere, P. D.; Anthonis, J. Incorporation of AUTOSAR in an Embedded Systems Development Process: A Case Study. In *2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications*; 2011; pp. 247–250.
20. Sangiovanni-Vincentelli, A.; Natale, M. D. Embedded System Design for Automotive Applications. *Computer* **2007**, *40*, 42–51, doi:10.1109/MC.2007.344.
21. Chakraborty, S.; Lukasiewicz, M.; Buckl, C.; Fahmy, S.; Chang, N.; Park, S.; Kim, Y.; Leteinturier, P.; Adlkofer, H. Embedded Systems and Software Challenges in Electric Vehicles. In *Proceedings of the Conference on Design, Automation and Test in Europe; DATE '12*; EDA Consortium: San Jose, CA, USA, 2012; pp. 424–429.
22. Sethi, P.; Sarangi, S. R. Internet of Things: Architectures, Protocols, and Applications Available online: <https://www.hindawi.com/journals/jece/2017/9324035/abs/> (accessed on Apr 16, 2018).
23. Pajares Redondo, J.; Prieto González, L.; García Guzman, J.; L. Boada, B.; Díaz, V. VEHIOt: Design and Evaluation of an IoT Architecture Based on Low-Cost Devices to Be Embedded in Production Vehicles. *Sensors* **2018**, *18*, 486, doi:10.3390/s18020486.
24. McMullen, S. A. H.; Henderson, T.; Ison, D. Embry-Riddle Aeronautical University multispectral sensor and data fusion laboratory: a model for distributed research and education. In *Next-Generation Spectroscopic Technologies X*; International Society for Optics and Photonics, 2017; Vol. 10210, p. 102100T.
25. Gomez-Cruz, J.; Nair, S.; Manjarrez-Hernandez, A.; Gavilanes-Parra, S.; Ascanio, G.; Escobedo, C. Cost-effective flow-through nanohole array-based biosensing platform for the label-free detection of uropathogenic E. coli in real time. *Biosens. Bioelectron.* **2018**, *106*, 105–110, doi:10.1016/j.bios.2018.01.055.
26. Ambrož, M. Raspberry Pi as a low-cost data acquisition system for human powered vehicles. *Measurement* **2017**, *100*, 7–18, doi:10.1016/j.measurement.2016.12.037.
27. Fog computing architectures for healthcare | Wireless performance and semantic opportunities | Journal of Information, Communication and Ethics in Society | Vol 14, No 4 Available online:

- <https://www.emeraldinsight.com/doi/abs/10.1108/JICES-05-2016-0014> (accessed on Apr 16, 2018).
28. Singh, H.; Yadav, G.; Mallaiah, R.; Joshi, P.; Joshi, V.; Kaur, R.; Bansal, S.; Brahmachari, S. K. iNICU – Integrated Neonatal Care Unit: Capturing Neonatal Journey in an Intelligent Data Way. *J. Med. Syst.* **2017**, *41*, 132, doi:10.1007/s10916-017-0774-8.
 29. Barik, R. K.; Dubey, H.; Samaddar, A. B.; Gupta, R. D.; Ray, P. K. FogGIS: Fog Computing for geospatial big data analytics. In *2016 IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON)*; 2016; pp. 613–618.
 30. Patel, S. M.; Kanawade, S. Y. Internet of Things Based Smart Home with Intel Edison. In *Proceedings of International Conference on Communication and Networks; Advances in Intelligent Systems and Computing*; Springer, Singapore, 2017; pp. 385–392 ISBN 978-981-10-2749-9.
 31. Mangeas, M.; Glaser, S.; Dolcemasclo, V. Neural networks estimation of truck static weights by fusing weight-in-motion data. In *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997)*; 2002; Vol. 1, pp. 456–462 vol.1.
 32. Gajdar, T.; Rudas, I.; Suda, Y. Neural network based estimation of friction coefficient of wheel and rail. In *Proceedings of IEEE International Conference on Intelligent Engineering Systems*; 1997; pp. 315–318.
 33. Darccedil; Akin, I.; Buuml; Akbaş, L. A neural network (NN) model to predict intersection crashes based upon driver, vehicle and roadway surface characteristics. *Sci. Res. Essays* **2010**, *5*, 2832–2836.
 34. Ozkurt, C.; Camci, F. Automatic Traffic Density Estimation and Vehicle Classification for Traffic Surveillance Systems Using Neural Networks. *Math. Comput. Appl.* **2009**, *14*, 187–196, doi:10.3390/mca14030187.
 35. Coninck, E. D.; Verbelen, T.; Vankeirsbilck, B.; Bohez, S.; Simoens, P.; Demeester, P.; Dhoedt, B. Distributed Neural Networks for Internet of Things: The Big-Little Approach. In *Internet of Things. IoT Infrastructures; Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*; Springer, Cham, 2015; pp. 484–492.
 36. Umakirthika, D.; Pushparani, P.; Rajkumar, M. V. Internet of Things in Vehicle Safety – Obstacle Detection and Alert System. *Int. J. Eng. Comput. Sci.* **2018**, *7*, 23540–23551, doi:10.18535/ijecs/v7i2.05.
 37. Razafimandimby, C.; Loscri, V.; Vegni, A. M. A Neural Network and IoT Based Scheme for Performance Assessment in Internet of Robotic Things. In *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*; 2016; pp. 241–246.
 38. VBOX Automotive - VBOX Available online: <https://vboxautomotive.co.uk/index.php/en> (accessed on Apr 16, 2018).
 39. Raspberry Pi 3 Model B Available online: <https://www.sparkfun.com/products/13825> (accessed on Apr 16, 2018).
 40. Industries, A. Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055 Available online: <https://www.adafruit.com/product/2472> (accessed on Apr 16, 2018).
 41. BNO055 Available online: https://www.bosch-sensortec.com/bst/products/all_products/bno055 (accessed on Apr 16, 2018).
 42. Intel® Edison Compute Module (IoT) Product Specifications Available online: <https://ark.intel.com/products/84572/Intel-Edison-Compute-Module-IoT> (accessed on Apr 16, 2018).
 43. SparkFun Block for Intel® Edison - 9 Degrees of Freedom - DEV-13033 - SparkFun Electronics Available online: <https://www.sparkfun.com/products/retired/13033> (accessed on Apr 16, 2018).
 44. Pieri, F.; Zambelli, C.; Nannini, A.; Olivo, P.; Saponara, S. Limits of sensing and storage electronic components for high-reliable and safety-critical automotive applications. In *2017 International Conference of Electrical and Electronic Technologies for Automotive*; 2017; pp. 1–7.
 45. FANN Available online: <http://leenissen.dk/fann/wp/> (accessed on Apr 16, 2018).

46. Zheng, J.; Qi, M.; Xiang, K.; Pang, M. IMU Performance Analysis for a Pedestrian Tracker. In *Intelligent Robotics and Applications*; Lecture Notes in Computer Science; Springer, Cham, 2017; pp. 494–504.
47. Herlihy, L.; Golen, E.; Reznik, L.; Lyshevski, S. E. Secure communication and signal processing in inertial navigation systems. In *2017 IEEE 37th International Conference on Electronics and Nanotechnology (ELNANO)*; 2017; pp. 414–419.
48. Parvis, M.; Grassini, S.; Angelini, E.; Scattareggia, P. Swimming symmetry assessment via multiple inertial measurements. In *2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*; 2016; pp. 1–6.
49. Using the GNU Compiler Collection (GCC): Optimize Options Available online: <https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html> (accessed on Apr 17, 2018).
50. *fann: Official github repository for Fast Artificial Neural Network Library (FANN)*; Fast Artificial Neural Network Library (FANN), 2018;