Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-27 (2019) 92–97

Experimental Autonomous Car Model with safety sensor in Wireless Network

Frederik Valocky * **Milos Orgon** * **Ina Fujdiak** **

* *Slovak University of Technology, Ilkovicova 3, Bratislava, 81107, Slovakia (e-mail: {valocky, orgon}@ut.fei.stuba.sk).*
** *Masaryk University, Zerotinovo nam. 617/9 Brno 601 77 Czech Republic (e-mail: {417345}@mail.muni.cz)*

Abstract: Autonomous vehicles are vehicles equipped with autonomous control systems that allow certain aspects of the control functions important for safe traffic to be controlled by the vehicle itself. At the same time, these cars are able to move from point A to point B separately in a defined environment and decide independently and adapt to unknown situations and a changing environment. These actions are autonomous cars capable of performing with minimal or no interference from the driver. This article is aimed at verifying the communication capabilities with data from safety sensors mounted on autonomous car between a control unit and car.

Keywords: Autonomous vehicles, Ultrasonic sensor, VANET, IoT, IoV

1. INTRODUCTION

Internet of Things (IoT) is already very well known and investigated phenomena with exponential growth in business, research and development Masek et al. (2016); Krivtsova et al. (2016); Papcun et al.. Nowadays, there are already several recognized sub-groups of IoT, such as Industrial IoT (supervisory SCADA systems and industrial control ICS systems) Gilchrist (2016); Lojka et al. (2016); Zolotová and Landryová (2005); Lojka and Zolotová (2014), Medical IoT Dimitrov (2016), IoT for Smart Cities (smart public lighning, optimized traffic management, or intelligent waste collection) Fujdiak et al. (2016b); Mlynek et al. (2015b); Fujdiak et al. (2016a, 2017), IoT for Smart Home Mlynek et al. (2016b), Wearable and mobile IoT Papcun et al. (2016); Arias et al. (2015); Zolotova et al. (2013), Smart Grid IoT (reliable and secure solutions, substation automation, circuit and failure indication, advanced universal metering and others) Fujdiak et al. (2016c); Raso et al. (2015); Fujdiak et al. (2015), Smart metering IoT (power line smart meters, radio and radio-mesh like networks, or generally wireless sensor networks for smart metering) Mlynek et al. (2015a,c, 2018, 2016a), Secure IoT (secure key management for IoT, end-to-end security, privacy, and other issues) Fujdiak et al. (2018a); Mihal' and Zolotová (2014); Fujdiak et al. (2018b), energy-efficient solutions and architectures Mocnej et al. (2018b,a), decentralized IoT implementations and infrastructures Mocnej et al. (2016), or Vehicle IoT Kaiwartya et al. (2016). This paper focuses on the last mentioned, the Vehicle IoT (VIoT), especially the area of autonomous vehicles and mobility.

The research Authors in Fazio et al. (2016, 2014) dealt with vehicular mobility predictions and admission control schemes in cellular networks. Next important issue of autonomous driving is data dissemination in cooperative vehicular networks Kurmis et al. (2015, 2014). Authors in Mehic et al. (2016) address a problem of the efficiency of routes in mobile Ad-Hoc networks, their approach to quality-link metrics is based on calculation of the lost probabilities of links by broadcasting probe packets.

From standard point of view, the National Highway Traffic Safety Administration (NHTSA), responsible for the operation of local roads and highways, defined National Highway Traffic Safety Administration and others (2013) five levels of vehicle automation. These levels symbolize the development of the driving assistants. A level 0 car does not offer the driver any driving assistants. The first-level car is equipped with driver-facilitating assistants, for example, anti-lock braking system (ABS), Electronic Stability Program (ESP), Anti-Slip Regulation (ASR) or Adaptive Cruise Control (ACC). The second-level car is fitted with a lane tracking system that keeps the car in the longitudinal direction. Third-level cars are represented by systems replacing the driver only when certain conditions are met, such as highway or clear ground road. Level 4 car can be labelled as a fully autonomous car that is capable of replacing the driver all the time.

This paper introduces the issue of Internet of Vehicles together with brief overview of state of the art. The main scope of the paper is to bring recent results from developing the autonomous car prototype for research purposes. The main purpose of prototype is move without any action executed by man. These results should help with future testing of technologies, methods and algorithms in the area of autonomous driving, remote vehicle control and general development in VIoT. The rest of this paper is organized as follows: Section II introduces our developed prototype of autonomous car model together with functional and hardware parts. Further, Section III provides description for experimental validation of the main parts - radar design and remote control system design. Finally, Section IV summarizes our findings and suggests future work.

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.12.739

2. AUTONOMOUS PROTOTYPE MODEL

2.1 Design and implementation of an autonomous vehicle

In order to verify the capability of an autonomous vehicle, it was necessary to create a miniature autonomous vehicle. The final prototype of our autonomous car is shown in Figure 1.

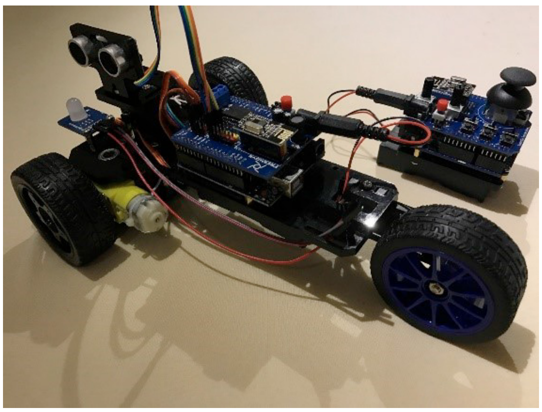

Fig. 1. Prototype of developed autonomous vehicle with remote control unit.

Among the mechanical part of our prototype, the main parts are autonomous car tower board and remote controller tower board (Figure 2 - part A), which create the autonomous intelligence of developed autonomous car. Autonomous car tower and remote controllers board tower use motherboard Arduino UNO (Figure 2 - part B, PCB1). Assembled printed circuit board (PCB) developed for the board tower of remote controller are shown in Figure 2 - part C (PCB2) and part D (PCB3). The PCB2 and PCB3 boards do not need to be connected by cables because they were preconditioned at the factory to connect all the pins from the motherboard directly to the Arduino Uno (PCB1) motherboard using the pins placed on the bottom of the PCBs into the openings on the motherboard. Subsequently, other PCBs already assigned to the operation or use of the vehicle could be connected to the PCB2 and PCB3 boards.

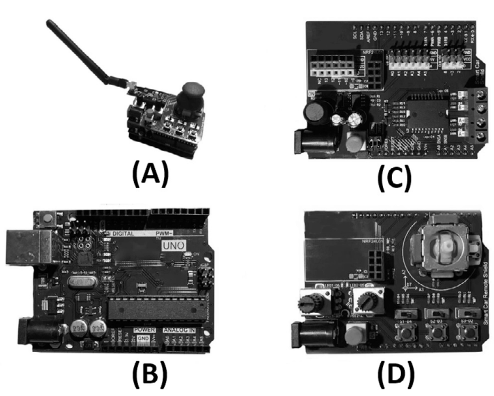

Fig. 2. Developed autonomous intelligence towers for autonomous car and remote controller.

The main mechanical components settings was as followed. The first servomotor used in the vehicle served to rotate the front axle with the wheel drive according to the position of the control element on the actuator and the second was switched on continuously to ensure movement of the ultrasonic sub module. A DC motors are a magnet mounted in a coil that converts a direct current to mechanical energy. The rotational speed of unidirectional motors can be varied over a wide range using either a changing supply voltage or current. It is important to properly connect the polarity of the source. Two sub modules with Wi-Fi technology from the PCBs nRF24L01+ series were tested in the design of the vehicle and remote controller. The first type with the built-in PCB3 antenna was powered by a lower voltage (3V). The second PCB4b was powered by a higher voltage with an external antenna. The Wi-Fi transmitter worked in the 2.4 GHz bandwidth. The frequency range was 2400 to 2525 MHz. One channel width for the nRF24L01+ board was 1 MHz, providing 125 possible channels with numbers ranging from 0 to 124. For the nRF24L01+ board, it was recommended by the manufacturer to use up to 25 channels. For detection, the HC-SR04 ultrasonic sensor PCB was used to measure the distance. The ultrasonic sensor worked on the principle of transmitting electromagnetic waves, which, after reflection, came back and were captured by the receiver. The HC-SR04 sub module used in the proposed vehicle was powered with a higher DC voltage of 5 V and a 15 mA current. A frequency of 40 Hz was used, with the HC-SR04 sub module transmitting a pulse every 10 s. The range of sensors ranged from 2 cm to 4 m and the radius of measurement was 15 degrees. The car is powered by two 18650 accumulators to provide enough power for the engines. The controller was powered by a 6LF22 battery by connecting it to the Arduino UNO motherboard. After the controller and vehicle were assembled, there was an error - the connection between the vehicle and the controller was impossible, and therefore the PCB2 with Wi-Fi sub module was powered by a lower voltage (3V) - nRF24L01+ replaced by a more powerful PCB with an external antenna nRF24L01 + (PCB4b) was powered by a higher voltage (5 V). After this change, it was already possible to establish a connection between the controller and the vehicle. For final testing the autonomous vehicle was loaded with several protectors as shown in Figure 3.

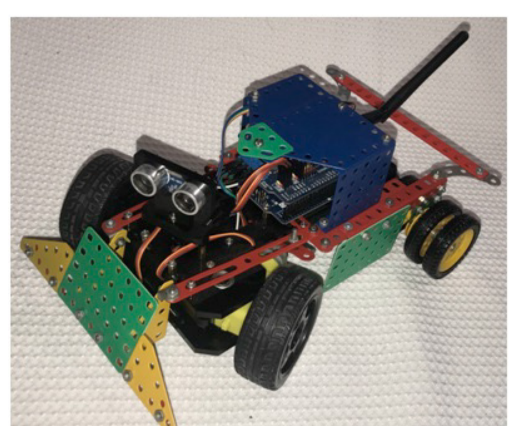

Fig. 3. Prototype of developed autonomous vehicle with loaded protectors.

3. EXPERIMENTAL MODEL VALIDATION

3.1 Control by camera

The main part of autonomous driving is executed by control unit with connected camera located on the roof of building. Camera is capturing real-time video. Control unit with connected camera have running script which locates actual position of autonomous car model. After control unit catch model position send command for next movement of car. The schema of autonomous movement system is shown in Figure.4

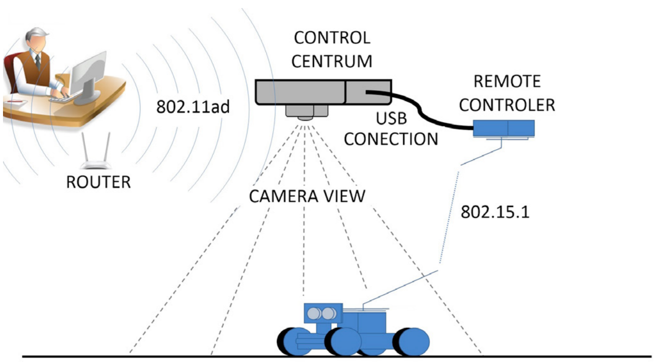

Fig. 4. Schema of locating position of autonomous vehicle and communication for sending control commands.

3.2 Radar validation

From many of capabilities for secondary safety system for autonomous car model we choose ultrasonic sensor. For this model is ultrasonic sensor sufficient. For bigger models must be more then one sensor and also more different type of sensors. Signal from ultrasonic sensors is shown in The Radar application. The Radar application was created in the freely available Processing 3 program by using the Java programming language. The Radar application was created in such a way that the data is received from a autonomous vehicle that must be wirelessly connected to the vehicle using the nRF24L01+ Wi-Fi module. On vehicle is located the ultrasonic sensor HC-SR04. The data from the remote controller is transmitted through the USB interface, one cable end is in the computer running the Radar application and the other end is plugged into the remote controller. When the application is started, a start screen appears on the computer screen. When the start screen is running, the computer searches all USB ports sequentially with COM1, COM2, and COMn, where n represents any USB port number on which the driver is connected.

When you find the USB port with connected remote controller, the Radar screen appears. If the error message "WARNING! Car is not connected" is displayed on the computer screen, you must check that the vehicle is powered or not. Whether the vehicle is switched on (switches on the PCB2 switch to active mode and whether the connection is active). The indication of power supply is connected or switching on/off button on the vehicle is carried out by means of the LEDs on the Arduino UNO and on the PCB2 is turn on. If the LEDs on the Arduino UNO motherboard are illuminated and the PCB2 is not, there is a power failure. When this error was detected, instead of 18650 batteries that were only connected to the PCB2, two 9V type batteries were used 6LF22-one for PCB2 power supply and the other for powering the UNO-PCB1 motherboard, but the connection was not able to be set up. In addition to the error message in the Radar application, a connection error on the controller was also indicated by the LED located on the LED3- D8, next to the pins to connect to the Wi-Fi module of exchanged modules.

When testing the Radar application, it was necessary to create a place with a moving obstacle and the need for a precise distance meter to compare the distance values obtained from the ultrasonic sensor. For comparison, the BOSCH DLE 70 Professional Laser Distance Meter (Figure 5) was used as professional reference measuring device. As a moving obstacle, the carton was used, which was moved to different distances. To make the measurement even more precise, the PCB2 has been disconnected from the automotive power supply for the servomotor controlling the rotation motion of the ultrasonic sensor and subsequently set to a direct position. Therefore, the measurement captured by the radar screen photograph, the points are located at -45°to + 45°, although they are within the maximum range of 15°, which means that focusing on the exact center of the obstacle would the points should be within the range of -7.5°to 7.5°. However, the Radar application counts the rotary motion of the sensor and therefore the points are evaluated in the 90°range.

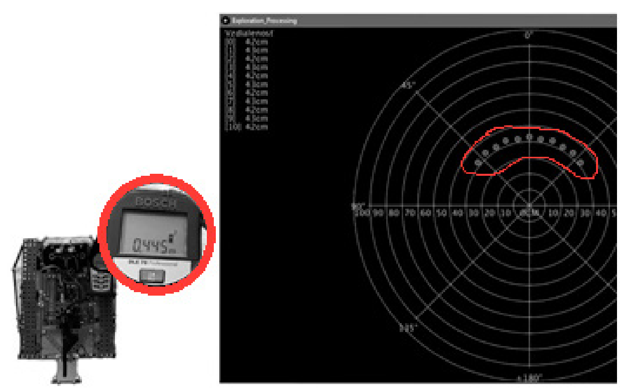

Fig. 5. Prototype of developed autonomous vehicle with loaded protectors.

To compare the data, measurements were made to detect a possible error of the radar distance. If the radar did not show all eleven points for the same distance, the arithmetic mean was calculated from the existing values. The measured values are shown in the following table.

Table 1. Comparative Experimental measurement of obstacle distance.

#	HC-SR04 [cm]	BOSCH DLE 70 [cm]
1	13.0	13.7
2	42.5	44.5
3	57.3	60.5
4	79.1	81.9
5	80.0	83.2

The radar application capable of measuring distance up to 100 cm in radius, but points on a circle determining the distance are only displayed up to a maximum distance of 79 cm, so that at 80 cm distance the radar points no longer appear. The average distance deviation was 2.38 cm, which corresponds to the distance from the sensor on the front side of the vehicle.

3.3 Vehicle control validation

To control a car in radar mode, it was necessary to create remote controller and vehicle codes that were then uploaded to the motherboards. The header contains the Serial Peripheral Interface (SPI) header, the servo motor and the RF24 communication modules. First, it was necessary to create a secure communication channel in which only the remote controller and the vehicle would communicate. The aim of such a solution was to exclude the possibility of unauthorized access by a potential attacker capable of controlling an autonomous vehicle. This problem has yet been resolved by a five-character password that may contain any characters. The password is part of the code so it cannot be changed after uploading to the motherboard. The same password must also include the remote controller code to be able to establish a connection to the vehicle. The part that creates secure communication is as follows:

Listing 1. Secure communication code example.

```
RF24 radio(9,10);
  //ID of pins CE=9 for standby mode and
  //CSN=10 for SPI commnuication
const byte adresa[6] = "Passw";
  //defined communiction address with remote
  //controller is coresponding
```

From the part of the code where ports, elements, and modules are set, a section was selected that opens the read and write channel. In this way, communication between the remote controller and the vehicle is finally created. The other parts of the code are designed to set the other elements and then write, measure, and store the necessary data from the Radar application and the servomotors needed to correctly control the vehicle. Part of the code opening the communication is as follows:

Listing 2. Session opening code example.

```
radio.begin();
  //inicialization RF24
radio.setPALevel(RF24_PA_LOW);
  // sets the power of the power amplifier
radio.setDataRate(RF24_1MBPS);
  // setting the data transfer rate
radio.setRetries(0, 15);
  // setting the number to be repeat
radio.openWritingPipe(adresa);
  // command for write
radio.openReadingPipe(1, adresa);
  // command for read
radio.startListening();
  // monitoring
```

These two parts of the code are required for both Arduino boards. For remote controller and also for the vehicle. The code for the remote controller is much easier because it only ensures the transmission of information and control data from the vehicle and the vehicle and the subsequent transfer to the computer via a USB cable. The code for the vehicle is more complex because it has to contain much more control information with informations of the pins to which the modules are connected (servomotor, DC motors, ultrasonic sensor). One servomotor must be controllable by the remote controller, and the other is set to 90°radius. Engines must have a defined speed (speed changes through the code because they are electromotors). Vehicle code must also be set to read data from an ultrasonic sensor.

When testing the vehicle's autonomous control, an obstacle course was created, which had the vehicle unobstructed to override the obstacle and continue in drive. In the event of an obstruction impossible, the vehicle also allowed backward movement and attempted to find a new way. In the code written for the stand-alone vehicle, the forward and backward speeds are given. These parameters are entered in the Vehicle Movement Code when using the Radar application, but the vehicle is in the autonomous mode not man-operated, so consider the vehicle's reaction time and the distance-based obstruction. For this purpose, different vehicle distances and speeds have been tested in order to be able to overcome the obstacle. It was not a simple task, because while reducing speed and increasing distance it was theoretically possible to find satisfactory speed and distance equivalent to autonomous control, but there was one serious problem. The vehicle, together with the batteries, has a certain weight and the vehicle's downshifting speed does not need to be moved at the required speed either. That was a critical point at which it was necessary to stop the reduction in speed. After longer-lasting testing, the optimal forward speed was determined to 140 and the backward speed to 95. The numbers shown merely represent a speed value in terms of control over a range of possible values from 0 to 255; values 140 and 95 are actually control information from a given range of values. The variable speed setting of the engines was then performed as follows:

Listing 3. Speed settings code example.

```
#define FORWARD LOW
  //LOW- negative voltage to motors - forward
#define BACKWARD HIGH
  //HIGH- positive voltage to motors - backward
const byte motorSpeed = 140;
  // motor speed (0-255)
```

Furthermore, it was necessary to define in the code the rotation angles of the servomotor through which the direction of the vehicle changes. These angles depended on the location of the obstacle. The aim was to allow the vehicle to adjust the angle that is needed to obviate the obstacle. It was taken into account that the maximum angle of the servomotor was 90°, so it was decided that the maximum angle to the left would be 135°and 45°to the right.

However, in connection with the rotation angle setting, another problem has emerged. After the conversion, it was found that the control electronics did not suffice at a certain speed of the vehicle to determine sufficiently quickly another obstacle, which was due to an impact on the obstacle. The vehicle was unable to measure and evaluate the distance (or to respond sufficiently to a rapidly approaching obstacle) when obstructing the obstacle, if the speed was too high. To circumvent the obstacle or move backwards depending on the distance and the rotation angle, the following condition was created in the code:

Listing 4. Backward and rotation code example.

```
if (distance < 20) {
  // distance in mm which you need to move backward
    if (angle < 90)
      // angle at which it is necessary to engage
     ctrlCar(90, BACKWARD, 95);
       // ctrlCar - command for car controll
    else
      ctrlCar(90, BACKWARD, 95);
  }
else if (distance < 50) {
  // distance in mm which you need to move forward
    if (angle < 90)
      // angle at which it moves forward
      ctrlCar(135, FORWARD, motorSpeed);
    else
      ctrlCar(45, FORWARD, motorSpeed);
  }
else {
  // no obstacle, goes straight ahead
    ctrlCar(90, FORWARD, motorSpeed);
```

In this context, it is still necessary to explain the ctrlCar command control function. This command was created to make writing easier to manage. The ctrlCar command has three parameters. The first parameter specifies the position of the servomotor (i.e., wheel positions). This parameter is an integer. This code represents a 90°precision center - so the vehicle moves straight in the direction of travel. The second parameter is defined at the beginning of the code and determines the polarity of the voltage to which the electric motors are powered, i.e., Whether the vehicle will move forward or backward - this parameter can only have two pre-defined options (forward / backward). The third parameter is an integer that determines speed of DC motors. The speed has an integer value of 95. When moving forward, the value of the variable at the beginning of the code is specified as motorSpeed (value 140). Other parts of the code provide for reading and reversing data from electric motors, servomotors, and ultrasonic sensors.

4. CONCLUSION

We summarized our latest results from developing the prototype for autonomous vehicles. The obtained results shows that miniature model might be a very promising way for testing the new technologies in this area. The most crucial part for future research should be focused on several topics: (i) precise navigation system, (ii) fast real-time obstacle detection (in the paper was introduced slow-speed detection, but more research in fast real-time detection must be done), (iii) security issue of the 802.11 channel (WiFi or cellular networks will be the future communication technology for autonomous cars, but these communication channels often come also with many security concerns and issues), and (iv) vehicle-to-vehicle communication model or vehicle-to-operator model.

ACKNOWLEDGEMENTS

The paper was partially supported by the Slovak Research and Development Agency, grant No. APVV-17-0190 and the Slovak Cultural Educational Grant Agency, grant No. 038STU-4/2018.

REFERENCES

Orlando Arias, Jacob Wurm, Khoa Hoang, and Yier Jin. Privacy and security in internet of things and wearable devices. *IEEE Transactions on Multi-Scale Computing Systems*, 1(2):99–109, 2015.

Dimiter V Dimitrov. Medical internet of things and big data in healthcare. *Healthcare informatics research*, 22 (3):156–163, 2016.

Peppino Fazio, Mauro Tropea, Cesare Sottile, Salvatore Marano, Miroslav Voznak, and Francesco Strangis. Mobility prediction in wireless cellular networks for the optimization of call admission control schemes. In *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–5. IEEE, 2014. doi: 10.1109/CCECE.2014.6900981.

Peppino Fazio, Mauro Tropea, Floriano De Rango, and Miroslav Voznak. Pattern prediction and passive bandwidth management for hand-over optimization in qos cellular networks with vehicular mobility. *IEEE Transactions on Mobile Computing*, 15(11):2809–2824, 2016. doi: 10.1109/TMC.2016.2516996.

Radek Fujdiak, Pavel Masek, Jiri Hosek, Petr Mlynek, and Jiri Misurec. Efficiency evaluation of different types of cryptography curves on low-power devices. In *2015 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 269–274. IEEE, 2015.

Radek Fujdiak, Pavel Masek, Petr Mlynek, Jiri Misurec, and Ammar Muthanna. Advanced optimization method for improving the urban traffic management. In *Proceedings of the 18th Conference of Open Innovations Association FRUCT*, pages 48–53. FRUCT Oy, 2016a.

Radek Fujdiak, Pavel Masek, Petr Mlynek, Jiri Misurec, and Ekaterina Olshannikova. Using genetic algorithm for advanced municipal waste collection in smart city. In *2016 10th International symposium on communication systems, networks and digital signal processing (CSNDSP)*, pages 1–6. IEEE, 2016b.

Radek Fujdiak, Jiri Misurec, Petr Mlynek, and Leonard Janer. Cryptograph key distribution with elliptic curve diffie-hellman algorithm in low-power devices for power grids. *Revue Roumaine des Sciences Techniques*, pages 84–88, 2016c.

Radek Fujdiak, Petr Mlynek, Jiri Misurec, and Jan Slacik. Simulation of intelligent public light system in smart city. In *2017 Progress In Electromagnetics Research Symposium-Spring (PIERS)*, pages 2515–2519. IEEE, 2017.

Radek Fujdiak, Petr Blazek, Konstantin Mikhaylov, Lukas Malina, Petr Mlynek, Jiri Misurec, and Vojtech Blazek. On track of sigfox confidentiality with end-to-end encryption. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, page 19. ACM, 2018a.

Radek Fujdiak, Petr Mlynek, Petr Blazek, Maros Barabas, and Pavel Mrnustik. Seeking the relation between performance and security in modern systems: metrics and measures. In *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pages 1–5. IEEE, 2018b.

Alasdair Gilchrist. *Industry 4.0: the industrial internet of things*. Apress, 2016.

Omprakash Kaiwartya, Abdul Hanan Abdullah, Yue Cao, Ayman Altameem, Mukesh Prasad, Chin-Teng Lin, and Xiulei Liu. Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects. *IEEE Access*, 4:5356–5373, 2016.

Irina Krivtsova, Ilya Lebedev, Mikhail Sukhoparov, Nurzhan Bazhayev, Igor Zikratov, Aleksandr Ometov, Sergey Andreev, Pavel Masek, Radek Fujdiak, and Jiri Hosek. Implementing a broadcast storm attack on a mission-critical wireless sensor network. In *International Conference on Wired/Wireless Internet Communication*, pages 297–308. Springer, 2016.

Mindaugas Kurmis, Dale Dzemydiene, Arunas Andziulis, Miroslav Voznak, Sergej Jakovlev, Zydrunas Lukosius, and Gediminas Gricius. Prediction based context data dissemination and storage model for cooperative vehicular networks. In *Nostradamus 2014: Prediction, Modeling and Analysis of Complex Systems*, pages 21–30. Springer, 2014. doi: 10.1007/978-3-319-07401-6_3.

Mindaugas Kurmis, Arunas Andziulis, Dale Dzemydiene, Sergej Jakovlev, Miroslav Voznak, and Gediminas Gricius. Cooperative context data acquisition and dissemination for situation identification in vehicular communication networks. *Wireless Personal Communications*, 85(1):49–62, 2015. doi: 10.1007/s11277-015-2727-1.

Tomáš Lojka and Iveta Zolotová. Improvement of human-plant interactivity via industrial cloud-based supervisory control and data acquisition system. In *IFIP International Conference on Advances in Production Management Systems*, pages 83–90. Springer, 2014.

Tomáš Lojka, Martin Miškuf, and Iveta Zolotová. Industrial iot gateway with machine learning for smart manufacturing. In *IFIP International Conference on Advances in Production Management Systems*, pages 759–766. Springer, 2016.

Pavel Masek, Radek Fujdiak, Krystof Zeman, Jiri Hosek, and Ammar Muthanna. Remote networking technology for iot: Cloud-based access for alljoyn-enabled devices. In *Proceedings of the 18th Conference of Open Innovations Association FRUCT*, pages 200–205. FRUCT Oy, 2016.

M Mehic, P Fazio, M Voznak, P Partila, D Komosny, J Tovarek, and Z Chmelikova. On using multiple routing metrics with destination sequenced distance vector protocol for multihop wireless ad hoc networks. In *Modeling and Simulation for Defense Systems and Applications XI*, volume 9848, page 98480F. International Society for Optics and Photonics, 2016. doi: 10.1117/12.2223671.

Roman Mihal' and Iveta Zolotová. Incidents, alarms and events in information and control systems. In *2014 IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 371–374. IEEE, 2014.

Petr Mlynek, Jiri Misurec, Radek Fujdiak, Zdenek Kolka, and Ladislav Pospichal. Heterogeneous networks for smart metering–power line and radio communication. *Elektronika ir Elektrotechnika*, 21(2):85–92, 2015a.

Petr Mlynek, Jiri Misurec, Zdenek Kolka, Jan Slacik, and Radek Fujdiak. Narrowband power line communication for smart metering and street lighting control. *IFAC-PapersOnLine*, 48(4):215–219, 2015b.

Petr Mlynek, Jiri Misurec, Michal Koutny, Radek Fujdiak, and Tomas Jedlicka. Analysis and experimental evaluation of power line transmission parameters for power line communication. *Measurement Science Review*, 15 (2):64–71, 2015c.

Petr Mlynek, Radek Fujdiak, Jiri Misurec, and Jan Slacik. Experimental measurements of noise influence on narrowband power line communication. In *2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 94–100. IEEE, 2016a.

Petr Mlynek, Zeynep Hasirci, Jiri Misurec, and Radek Fujdiak. Analysis of channel transfer functions in power line communication system for smart metering and home area network. *Advances in Electrical and Computer Engineering*, 16(4):51–56, 2016b.

Petr Mlynek, Jiri Misurec, Petr Toman, Pavel Silhavy, Radek Fujdiak, Jan Slacik, Zeynep Hasirci, and Konstantin Samouylov. Performance testing and methodology for evaluation of power line communication. *Elektronika ir Elektrotechnika*, 24(3):88–95, 2018.

Jozef Mocnej, Tomáš Lojka, and Iveta Zolotová. Using information entropy in smart sensors for decentralized data acquisition architecture. In *2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 47–50. IEEE, 2016.

Jozef Mocnej, Martin Miškuf, Peter Papcun, and Iveta Zolotová. Impact of edge computing paradigm on energy consumption in iot. *IFAC-PapersOnLine*, 51(6):162–167, 2018a.

Jozef Mocnej, Winston KG Seah, Adrian Pekar, and Iveta Zolotova. Decentralised iot architecture for efficient resources utilisation. *IFAC-PapersOnLine*, 51(6):168–173, 2018b.

National Highway Traffic Safety Administration and others. Preliminary statement of policy concerning automated vehicles. *Washington, DC*, pages 1–14, 2013.

Peter Papcun, Erik Kajati, Dominika Cupkova, Jozef Mocnej, Martin Miskuf, and Iveta Zolotova. Edge-enabled iot gateway criteria selection and evaluation. *Concurrency and Computation: Practice and Experience*, page e5219.

Peter Papcun, Iveta Zolotova, and Karim Tafsi. Control and teleoperation of robot khepera via android mobile device through bluetooth and wifi. *IFAC-PapersOnLine*, 49(25):188–193, 2016.

Ondrej Raso, Petr Mlynek, Radek Fujdiak, Ladislav Pospichal, and Pavel Kubicek. Implementation of elliptic curve diffie hellman in ultra-low power microcontroller. In *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*, pages 662–666. IEEE, 2015.

I Zolotova, L Laciňák, and T Lojka. Architecture for a universal mobile communication module. In *2013 IEEE 11th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 61–64. IEEE, 2013.

Iveta Zolotová and Lenka Landryová. Knowledge model integrated in scada/hmi system for failureprocess prediction. *WSEAS Transactions on Circuits and Systems*, 4(4):309–318, 2005.