International Journal
Information Technology Applications (ITA)

Volume 8, Number 2, December 2019

FUZZY CONTROLLER DESIGN AND CO-SIMULATION OF AUTOMATIC PARKING SYSTEM

Ján Cigánek

Abstract:

The aim of the paper is to propose a methodology control of parking process using fuzzy logic. The simplified vehicle, in which the design will be realized, is made in MSC Adams environment. This created vehicle model is afterwards exported to Matlab Simulink. In this environment a design is realized of the vehicle motion control through co-simulation using fuzzy controller. The design is tested and implemented for the control of the parking process in different parking situations, specifically for perpendicular and parallel parking.

Keywords:

Co-simulation, fuzzy controller, Matlab Simulink, MSC Adams, parking.

ACM Computing Classification System:

Computing methodologies, modeling and simulation, model development and analysis,
modeling methodologies.

◣ Introduction

Presently, the car is one of the most widespread transport vehicles. The automotive industry produces an incredible amount of cars every year. Their number is growing not only on public roads, but also in places reserved for car parking. Therefore, it is often the case that parking in places that are almost full is difficult and in many cases impossible for a person.

Parking assistant or automatic parking is no longer a novelty. Many car producers have this system and install it in their vehicles, either as an option or as part of the basic equipment of the car. It is a good help by parking in confined conditions.

The parking assistant is a system that facilitates parking of cars in both lateral (perpendicular, transverse) and longitudinal (parallel) parking. There are two types of parking assistants: those that allow manual steering of the vehicle by parking, or those that offer automatic parking.

The parking assist system consists of a set of ultrasonic sensors that are evenly distributed in both the front and rear part of the vehicle. The total number of sensors can be six or eight to twelve. Sensor data are processed in the control unit, which contains an algorithm for evaluating this data, and in the case of an automatic parking system, the algorithm is also extended to control action. The processed and evaluated data from the control unit can be presented to the driver on the display or other display units, along with an acoustic warning. The closer the vehicle is to the obstacle, the repetition frequency of the acoustic signal increases until it reaches a continuous tone, which indicates a very small distance from the obstacle.

The systems can also use a camera system to display the exact situation around the rear of the vehicle.

In automatic parking, the car itself has to make many intelligent decisions in a short time. Sensors have to find a suitable parking place. They can determine the length of free space, evaluate whether the vehicle can fit there, and send information to the control unit to calculate the correct trajectory of the parking maneuver. The best system needs sufficient parking space 80 cm larger than the length of the vehicle [1].

The first types of parking assist supported only the possibility of a longitudinal parking maneuver and allowed parking on the right side only. Presently, the systems also support cross parking and also allow parking on the left. Therefore, before starting to look for a parking space, the driver has a choice of what kind of parking and on which side of the vehicle he is interested. The maximum vehicle speed when searching for a parking space is up to 35 km / h. All data are processed, evaluated and displayed.

When searching for a parking space, it is necessary to activate the Park Assist system with the button and turn on the turn signal to the side where the driver wants to park. After finding a suitable parking space, the driver stops and engages reverse gear as instructed on the display. The driver only controls the vehicle speed when parking. The system evaluates the situation in front of and behind the vehicle and automatically rotates the steering wheel. When properly parked, the system shuts down automatically.

1 Vehicle Control

The defined movement of the vehicle model will be represented by the trajectory of the sensed point located in the center of the rear track of the vehicle model. At this point, the beginning of the XY coordinate system will also be here.

The vehicle model will move at a constant speed of $v = 0.277\ ms^{-1}$ throughout the parking maneuver. The generated reference trajectory will be compared to the actual trajectory that the vehicle model is currently executing.

Based on the deviation between these two trajectories, the steering intervention will be evaluated - representing the moment of force that will act on the steering mechanism of the vehicle model. This moment causes changes in the steering wheel angles of the vehicle model. In this way, it will be possible to control and correct the direction of movement of the vehicle model according to the reference trajectory of the sensed point.

The most widespread way to steer the direction of cars is Ackerman's steering. It is used the trapezoidal steering (Fig.1) to meet Ackerman's condition, wherein the steering arms together with the connecting rod are trapezoidal in shape [3].

After extending the front axle steering arms (*d* in Fig.1), the two arms are joined at one point - in the middle of the rear axle track. The angle between the arm and the wheel is called the Ackerman angle (β in Fig.1).

For the calculation of the Ackerman angle β is used the following equation:

$$\beta = arctg\ \left(\frac{A}{2\,L}\right) \tag{1}$$

where *A* is is front track gauge (mm) and *L* is wheelbase (mm).

The Ackerman steering condition: the center of rotation has to lie on the extended axle of the rear axle, as shown in (Fig.2). Fulfillment of this condition is an essential prerequisite for the proper rolling of steered wheels [3].

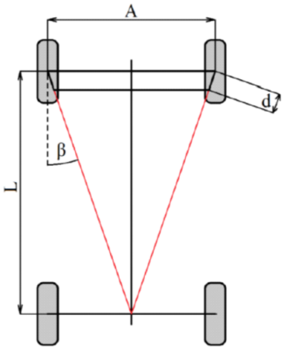

Fig.1. Creating a trapezoidal steering.

The relation between the angle alignment of the outer wheel δ_{OUT} and the angle alignment of the inner wheel δ_{IN} is:

$$cotg\ \delta_{OUT} - cotg\ \delta_{IN} = \frac{A}{L} \tag{2}$$

The equations to calculate the angle alignment of the inner wheel δ_{IN} and the angle of the outer wheel δ_{OUT}:

$$\delta_{IN} = arccotg\left(\frac{R-\frac{A}{2}}{L}\right) \tag{3}$$

$$\delta_{OUT} = arccotg\left(\frac{R+\frac{A}{2}}{L}\right) \tag{4}$$

where R is the turning radius (mm).

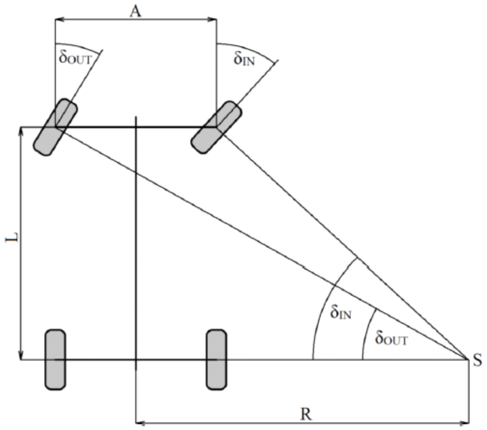

Fig.2. Angles of the wheels alignment at Ackerman steering.

Using the Pythagorean theorem, it is possible to derive the formula for calculating the turning radius of the vehicle R from the triangle in (Fig.3):

$$(L+B)^2+\left(R+\frac{A}{2}\right)^2=R_0^2 \tag{5}$$

leading to:

$$R=\sqrt{R_0^2-(L+B)^2}-\frac{A}{2} \tag{6}$$

where B is distance (mm) between the front axle and the point where turning vehicle describes the circle with the largest radius with the maximum possible front-wheel turning. R_0 is contour turning radius of the vehicle.

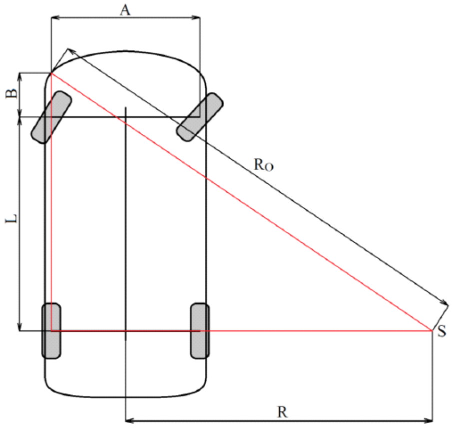

Fig.3. Angles of the wheels alignment at Ackerman steering.

2 Vehicle Modeling

To design a simplified vehicle model in the MSC Adams environment, the BMW 3 Series 320i has been selected. In order to realize the vehicle model, it was necessary to look for the dimensional and technical parameters of this vehicle, which are further used in the design of the steering wheels and trajectory control of the vehicle model. (Tab.1) shows the technical parameters necessary when designing a vehicle model [2].

Using equations (1 – 6) it is possible to calculate values of the Ackerman angle β, the turning radius R, the angle alignment of the inner wheel δ_{IN} and the angle of the outer wheel δ_{OUT}: $\beta=15.36°;R=3\,802.3\text{ mm};\ \delta_{IN}=42.84°;\delta_{OUT}=31.56°$.

The calculated angle alignments of both wheels are maximal and are based on the geometrical and driving characteristics of the vehicle. Their values are important when designing the steering wheels of the vehicle's front axle to avoid inaccurate steering caused by exceeding the maximum feasible steering angles of the front wheels. BMW models have a 50:50 front and rear axle load, it means the center of gravity is located in the center of the wheelbase of the vehicle.

Table 1. Technical parameters of the vehicle.

Length	4 633 mm
Width	2 031 mm
Height	1 429 mm
Curb weight	1 475 kg
Wheelbase *L*	2 810 mm
Front track gauge *A*	1 544 mm
Rear track gauge	1 583 mm
Contour turning diameter $2.R_0$	11 300 mm
Distance *B*	506.35 mm
Tire size	205/60 R16 92 H

Using the vehicle parameters (Tab.1) with the proposed front wheel steering control and the identified center of gravity, the vehicle model was built in MSC Adams (Fig.4).

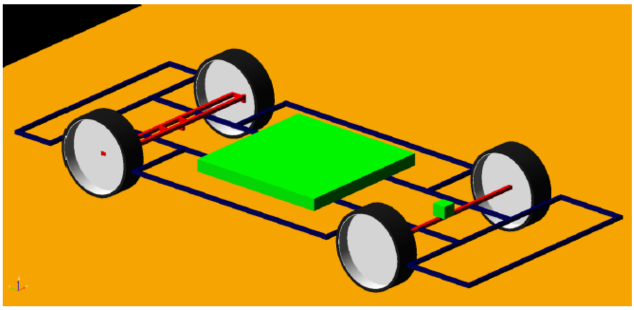

Fig.4. Vehicle model.

The vehicle model consists of several elements with major landmarks. One such point is called *Marker*. Individual elements of the vehicle model are its three basic parts: vehicle frame, wheels with its mountings and two additional weight bodies. These elements are interconnected by fixed or rotary coupling. Each coupling is defined between two elements.

The front wheel mounting elements also form the trapezoidal steering of the front axle and its individual elements are interconnected by rotary couplings to each other, allowing the front wheels to turn. The wheel mounting elements are fixed to the frame by fixed couplings. All wheels are attached to the wheel mounting elements by rotary couplings, allowing the wheels to rotate about their axis.

The vehicle model is located on a solid pad (dark yellow in Fig.4). A contact-type coupling is defined between the pad and each wheel, which ensures that all wheels roll on this pad.

In order to prevent possible exceeding of the maximum permissible angle alignments of the front, a mechanical fuse is also included in the steering mechanism of the front axle. The body marked PART_32 is fixedly attached to the trapezoid arm of the front wheel mount (PART_4), and which is fixed to the frame. The body marked PART_31 is fixedly mounted on the movable trapezoid arm of the front wheel mount (PART_7). A contact is formed between the PART_31 and PART_32 bodies. After the deflection of the arm PART_7, the body PART_31 is also deflected with it. After an ever-increasing deflection angle, it encounters an obstacle in the form of a body PART_32. The principle of the mechanism is shown in (Fig.5).

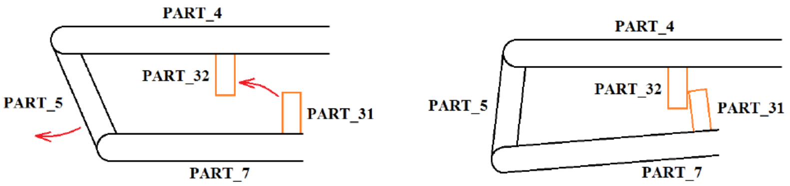

Fig.5. Mechanical safety device against exceeding the maximum steering angles.

The third part of the vehicle model consists of two bodies (green in Fig.4). They serve as an additional weight so that the mass of the vehicle model matches the curb weight *m = 1475 kg*. The second function is to balance the vehicle model so that the center of gravity is located in the center of the wheelbase.

3 Co-Simulation

The first important step is the definition of sensed and controlled variables in MSC Adams. Sensed variables are outputs that can be either informative or directly affect the movement control of the vehicle model. Controlled variables influence the vehicle model control and represent inputs.

The output variables of the system are:

- the position of the point in the X-axis direction,
- the position of the point in the Y-axis direction,
- the angle alignment of the left wheel and
- the angle alignment of the right wheel.

The first two output variables are used to control the movement of the vehicle model. The left wheel angle and the right wheel angle are used for informative monitoring and to control the maximum angle values while driving. The input variable is the moment of force applied to the left movable steering arm of the vehicle, the magnitude of the moment affects the angle alignment.

To export a vehicle model from MSC Adams to Matlab Simulink, it is necessary to create corresponding system state variables for all input and output variables that will be used to control the movement of the vehicle model. The individual variables used to control the movement of the vehicle model, together with their corresponding system state variables, are shown in (Tab.2).

Since the output variables do not directly affect the left and right angle alignment of the vehicle model, system state variables will not be created for these variables. However, their value can be monitored on the created meters in the MSC Adams environment.

Table 2. System state variables.

Moment	M
Position in the X-axis	X
Position in the Y-axis	Y

After defining all system status variables and all input and output variables, the assembled vehicle model is exported from the MSC Adams environment to the Matlab Simulink environment, in which the motion control of the vehicle model will be implemented using the fuzzy controller.

An important step by creating the control scheme is the design of a methodology for evaluating and comparing the reference trajectory with the actual trajectory. For the evaluation of the relationship between the position of the sensing point in the X-axis direction and the position of the sensing point in the Y-axis direction, the conversion to the angle φ is used. This angle is defined between the time changes in the position of the sensing point in both directions of the XY coordinate system, as shown in (Fig.6).

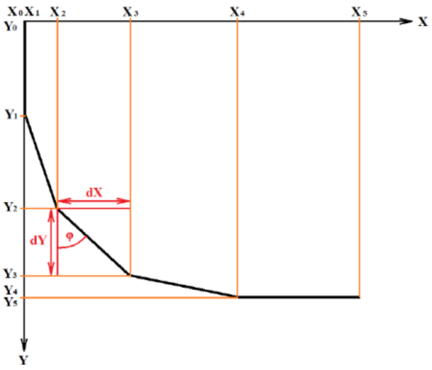

Fig.6. Conversion of XY coordinates to angle φ.

The calculation of the angle φ between changes in the position of the sensing point in the XY coordinate system is given by equation (7):

$$\varphi = arctg\left(\frac{dX}{dY}\right) \tag{7}$$

where *dX* is the time change of position in the X-axis direction and *dY* is the time change of position in the Y-axis direction. The calculation of *dX* and *dY* are given by the relations:

$$dX = X_A - X_P;\; dY = Y_A - Y_P \tag{8}$$

where X_A, Y_A are the current position values in the X-axis and Y-axis direction, X_P and Y_P are previous position values in the X-axis and Y-axis direction.

The angle φ between the time changes of the position of the sensed point in the XY coordinate system is calculated from the reference trajectory data as well as from the actual trajectory data. A simplified block diagram used for co-simulation and motion control of the vehicle model in the Matlab Simulink environment is shown in (Fig.7).

Two *Lookup Table* blocks contain time series of reference trajectory data in X-axis and Y-axis separately. The function, which is located in *Interpreted MATLAB Function* block, recalculates the time changes in the position of a point in the XY coordinate system from the reference trajectory to the angle φ using equations (7) and (8). The output values from the *adams_sub* block are the actual measured position data in the X-axis and Y-axis directions. This data enters the *Interpreted MATLAB Function1* block to calculate the angle φ for the current position change of the vehicle model. The calculated angles from both blocks are compared in the sum block with the output variable representing a control deviation *e*. The *Derivative* block calculates the derivative of the control deviation.

The control deviation between the angles and its derivative enter the *Fuzzy Logic Controller* block, where they are processed and evaluated. The output of this block is the moment *M* which controls the wheels steering of the vehicle model and corrects its movement according to the reference trajectory.

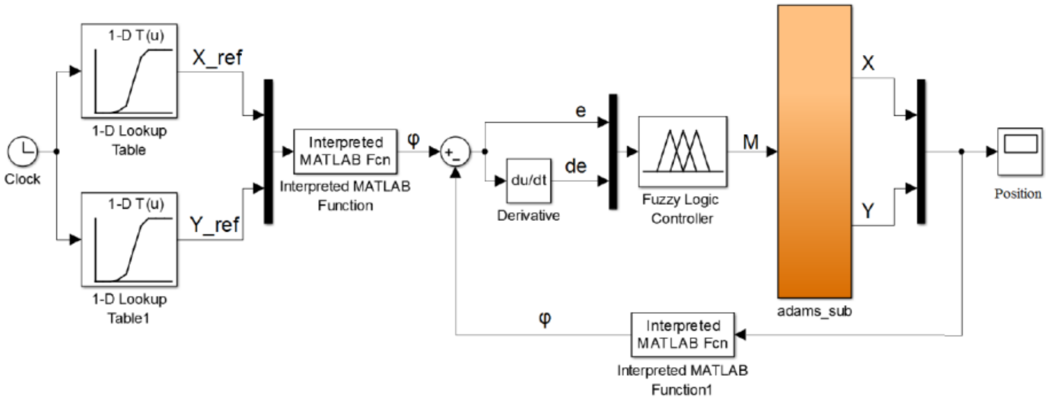

Fig.7. Simplified block diagram for co-simulation.

Design of fuzzy controller in Matlab Simulink environment is realized by *Fuzzy Logic Toolbox*. Typing the command “fuzzy” in the *Command Window* opens *FIS Editor* for fuzzy controller design. In this editor, it is possible to choose the type of fuzzy controller (Mamdani or Sugeno); to add controller inputs, controller outputs; to specify boundaries of individual input and output variables; to change numbers and types of membership functions; to set methods for implication, aggregation and defuzzification; to create rules; and to save the proposed fuzzy controller.

The proposed fuzzy controller of Mamdani type will enter two input variables: *e* represents the control deviation of angle φ with 5 membership functions (MFs) of triangular shape in range <-3°; 3°>; *de* represents the derivative of the control deviation with 3 MFs in range <-3°/sec; 3°/sec>. This fuzzy controller contains only 1 output variable *M* representing the moment with 7 MFs in range <-15 Nm; 15 Nm>.

The relations between the output variable *M* and the input variables *e* and *de* are determined by the rules. The most common way to create fuzzy rules is by IF - THEN. A fuzzy rule consists of two parts: the first is the conditional part of the rule (antecedent – consists of a combination of input variables) and the second one is the conclusion part of the rule (consequent – is made up of an output variable) [4]. All rules of proposed fuzzy controller are shown in (Tab.3).

Table 3. Rules of proposed fuzzy controller.

e / de	**Z**	**N**	**K**
ZV	KV	K	KM
ZM	K	KM	N
N	KM	N	ZM
KM	N	ZM	Z
KV	ZM	Z	ZV

Two parking situations were created based on the parameters of the designed vehicle model. Specifically, this concerns the situation of lateral (perpendicular, transverse) and longitudinal (parallel) parking. In both parking situations, vehicle model control was applied using the designed fuzzy controller.

In both parking situations, foreign bodies are added to the vehicle model to define a parking space for a particular parking situation. Subsequently, trajectories for movement of the sensing point of the vehicle model were created for both situations.

Due to the constant rearward movement of the vehicle model, both parking situations are designed and adapted so that the vehicle model can park reversely in a defined parking space based on one trajectory of the sensing point.

For the transverse (perpendicular) parking situation, the corresponding reference trajectory for vehicle movement is used (blue in Fig.8a). The length of the co-simulation was 23 seconds. After simulation, the trajectory of movement was recorded by the sensing point of the vehicle model. A comparison of the reference trajectory and the measured controlled trajectory of the vehicle model is shown in (Fig.8a).

Similarly, for the longitudinal (parallel) parking situation, the corresponding reference trajectory for vehicle movement is used (blue in Fig. 8b). The length of the co-simulation was 30 seconds. A comparison of the reference trajectory and the controlled trajectory of the vehicle model is shown in (Fig.8b).

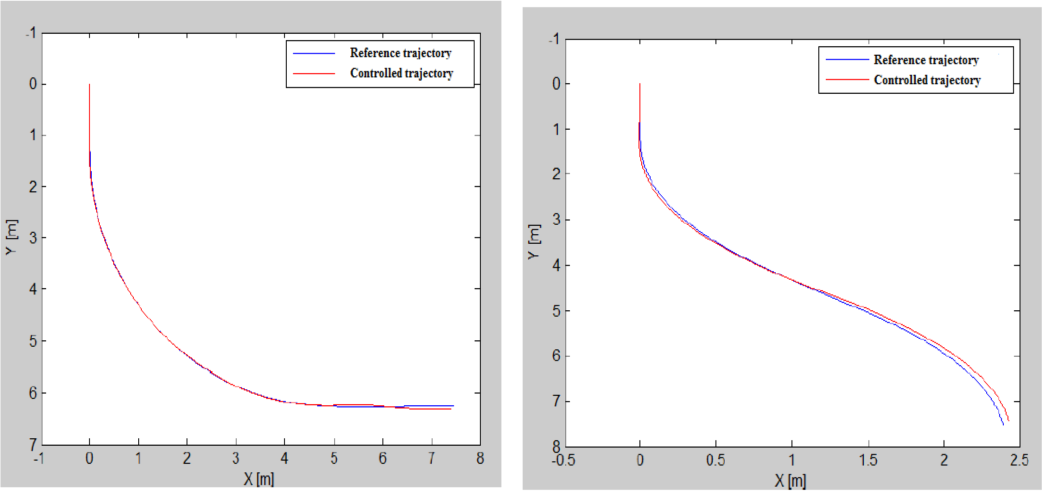

Fig.8. Comparison of reference and controlled trajectory: a) Transverse, b) Longitudinal.

It can be observed that in a transverse parking situation, the controlled trajectory of the vehicle model is almost identical to the reference trajectory. Vehicle model position during transverse parking maneuver is shown in (Fig.9a). Red bodies define a 2.5 x 5 meter parking space between them.

Similarly, in a longitudinal parking situation, the controlled trajectory of the vehicle model is very similar to the reference trajectory. Vehicle model position during longitudinal parking maneuver is shown in (Fig.9b). Red bodies define a 2.5 x 6.5 meter parking space.

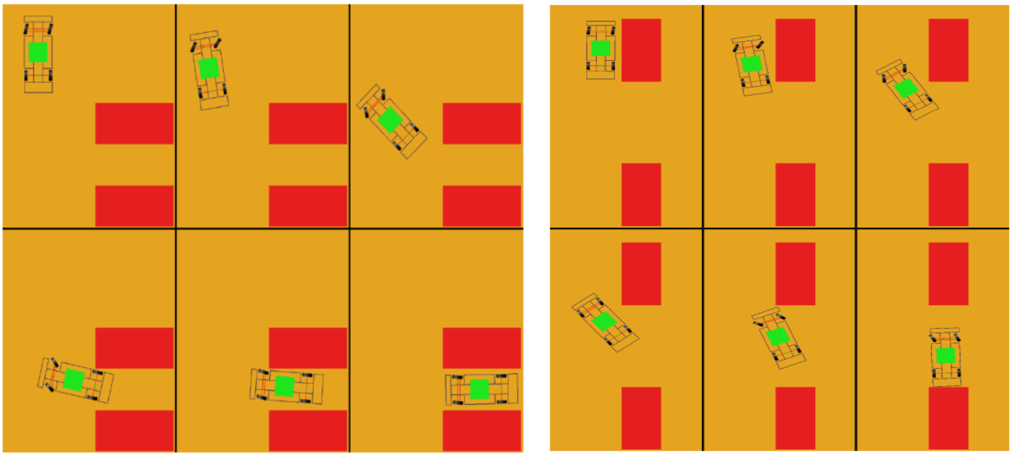

Fig.9. Positions during parking maneuver: a) Transverse, b) Longitudinal.

Conclusion

The aim of this work was to propose a methodology of vehicle parking process control using fuzzy logic and subsequently to verify it for different spatial situations. The method used to control the movement of the virtual vehicle model, which was designed in the MSC Adams environment, was based on the transformation of the X and Y coordinates of the sensed point of the vehicle model to an angle between the time changes of this point in the XY coordinate system. The motion control model was implemented in Matlab Simulink.

The proposed control was applied to two spatial parking situations in the form of transverse and longitudinal parking maneuver. From the achieved results, as well as from the graphical interpretation, it can be concluded that the proposed fuzzy controller is suitable for given parking situations.

Acknowledgement

This paper was supported by the Slovak Grant Agencies VEGA 1/0819/17, and by the Scientific Grants APVV-17-0190 and SEMOD-79-2/2019.

References

[1] Andrejčák, T. (2010). *Parking stress will be a thing of the past* (In Slovak) http://auto.pravda.sk/magazin/clanok/3837-stres-z-parkovania-bude-minulostou/

[2] Mihalik, M. (2018). *BMW is developing new assistance systems* (In Slovak) http://www.automagazin.sk/bmw-vyvija-nove-asistencne-systemy/

[3] Sajdl, J. (2020). *Ackermann condition* (In Czech) http://www.autolexicon.net/cs/articles/ackermannova-podminka/

[4] Ciganek, J., Noge, F. (2014). *Co-simulation of mechatronic system using fuzzy controller*. In Int. conference Cybernetics and informatics 2014. Oščadnica, Slovak republic, 5.-8. 2. 2014. Bratislava: SSKI, s. 6. ISBN 978-80-227-4122-4.

Authors

Ing. Ján Cigánek, PhD.
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
jan.ciganek@stuba.sk

He was born in 1981 in Malacky, Slovakia. He received the diploma and PhD. degree in Automatic Control from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology (FEI STU) in Bratislava, in 2005 and 2010, respectively. He is now Assistant Professor at Institute of Automotive Mechatronics FEI STU in Bratislava. His research interests include optimization, robust control design, computational tools, SCADA systems, big data, and hybrid systems.