

Determining the Embedded Key Performance Indicator (KPI), based on a Fuzzy FxLMS Algorithm

Péter Pál Bakucz, József Zoltán Szabó

Bánki Donát Faculty of Mechanical and Safety Engineering, Óbuda University,
Institute of Mechatronics and Vehicle Engineering
Népszínház utca 8, 1081 Budapest, Hungary
szabo.jozsef@bgk.uni-obuda.hu, bakucz.peter@bgk.uni-obuda.hu

Abstract: In this work, we present the evaluation of the algorithmic performance of perception, achieved by a video sensor, within the framework of the Highly Automated Driving (HAD) project of the Óbuda University. We process the quality of the real-time uncertainty propagation in the embedded environment. The perception software could be characterized by algorithmic key performance indicators (KPIs) which is the measurable metric of the video sequences uncertainty. Based on endurance runs, for the real-time performance determination of the algorithms, an adaptive filter structure could be approximated. The employed performance architecture is a fuzzy-filtered (F-Fx) version of the LMS (Least Mean Square) algorithm in the embedded autonomous driving software platform. The F-FxLMS model is designed for the prediction of the embedded real-time filter outputs. The key performance indicator is related to the adaptation number of the F-FxLMS filter.

Keywords: Uncertainty Analysis; Fuzzy Filter; Fuzzy FxLMS (F-FxLMS) Algorithm; Highly Autonomous Driving; Software Performance

1 Introduction

Determining embedded KPIs (key performance indicators) by filtered LMS algorithms, with periodic and deterministic sensor *reference*, are often the subject of the measuring and managing the quality of platform design in the highly automated driving (HAD) software development process [1].

The two primary *reasons* of the periodic and deterministic references are: the embedded microcontroller environment simplification requirements, and it is generally easier to find a good reference signal to model them.

In addition to these reference signals, frequently stochastic signals are used in real autonomous driving projects today. The algorithms are mathematically classified into filtered LMS algorithm sequences [11]. However, based on highly autonomous driving (HAD) research at the University of Óbuda, we considered that the “classical” filtered LMS algorithm should be supplemented with fuzzy features (F-FxLMS) in order to facilitate the modeling of uncertainty propagation [2] [3].

In addition, the main reason for fuzzification [8] is the basic issue to be found in numerical analysis \rightarrow when applying algorithms using classical deterministic references, there may be cases whose convergence has not been proved, which is called non-Wiener effects in the literature [9].

In previous work, which were presented at the SISY conference in 2019 [4], we showed in outline the steps of determining the software performance that can be used in the HAD project of the Óbuda University. Based on this, we present now a detailed analysis of the F-FxLMS algorithm and the test results performed with a vehicle with video sequences.

The essence of the adaptive filtering algorithm for determining embedded KPIs is based on the interpretation of the *transfer function* (W function) between different inputs and outputs.

Generally, to determine the embedded KPIs as a transfer function must be defined:

input *objects* (see Fig. 1) of the transfer W function of the F-FxLMS filter (in Óbuda University HAD Software see Fig. 1 defined by the Generation 5 video signals [11]), the environmental, digital signal processing *parameters* (period sampling), as well as uncertainty *values* on the *input* side [5].

At the same time, the *outputs* are the detected objects via object lists like object center, object width, and length. These parameters of the signal processing chain, are covering areas of the transfer function whose size far exceeds the unit, with the consequence that the number of interfering main pulses may increase [6].

The purpose of embedded filter-based performance analysis, is to define the maximum amount of transfer W function based on algorithmic uncertainty [7].

Determining the real-time filtering parameters for the optimal F-FxLMS illustrates, that it cannot be used to analyze stationary signals in specific frequency ranges. However, embedded considerations are needed to determine the duration of the entire system (in fuzzy filtering: via the neuro-estimated secondary path). These runtime additions extend the interpretation of the maximum stable filter parameter when viewing a specific frequency range.

The Matlab System Identification Toolbox [10] was recently released for the embedded, real-time analysis, of the classical FxLMS filtering and its filter behavior for autonomous driving projects, as approved for given microcontrollers.

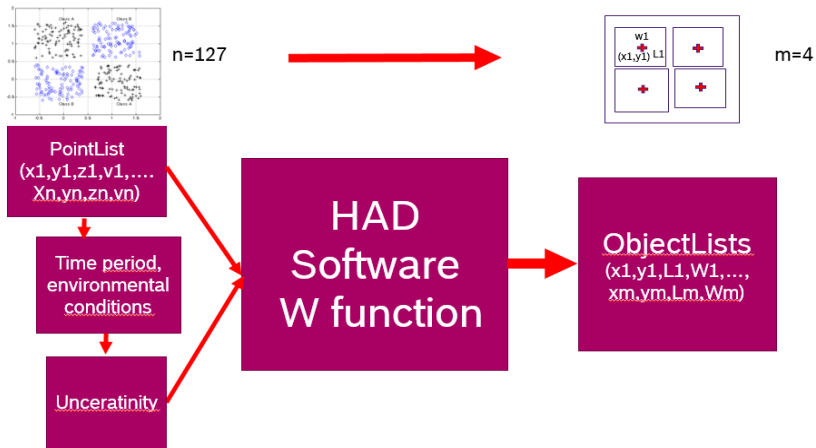


Figure 1

Óbuda University Human Autonomous Driving (HAD) software performance based on the input list (here $n=127$ video detected objects (points)) the time period, diverse environmental conditions, and the output (vehicle object lists; $m=4$, with centrum of the object x,y , and the width W and length L) (based on [14])

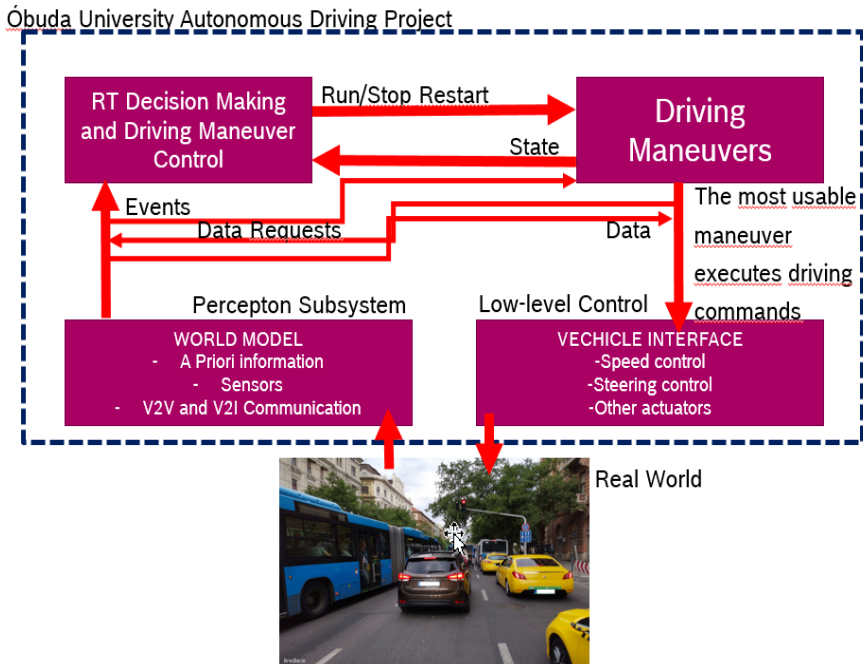


Figure 2

Autonomous Driving Control Application. The Óbuda University Autonomous Driving Project is engaged with the Perception Subsystem (video) software key performance indicators, KPIs [14].

In this work, we examine one of the software modules of the Software Sensing Subsystem of the Autonomous Driving Project of the Óbuda University and test our results in the real world as well in the form of testing with specific vehicles (Figures 3, 4). This software module is responsible for the sensor fusion detected on the video and radar channel, the “Perception” subclass.

To determine the embedded performance of the software module, we examined the analytical form of the W transfer function and the analysis of simple secondary sections with a clear delay. In possession of these, the maximum filter parameter of the F-FxLMS algorithm can be determined. Analysis of the poles of the W function provides data for the performance of the software module. This pole is related to the secondary path of the filter and the filter parameter and can be analytically expressed as required for embedded systems by maximizing the filter coefficients.

The application of the algorithm in the form of real-time measurements was performed within the framework of the Autonomous Driving Project of the University of Óbuda.

Our work presents in Sec. II, the basics of classical and fuzzy filtered (F-Fx) LMS algorithm. Afterward, in Sec. III, the designed procedure for autonomous driving software performance is presented. Sec. IV, provides the first results and Sec. V offers the Conclusions for this work.

Figure 3

Function structure of Óbuda University Autonomous Driving Software Sensor C++ system function component to be controlled by Fuzzy-FxLMS system [14]

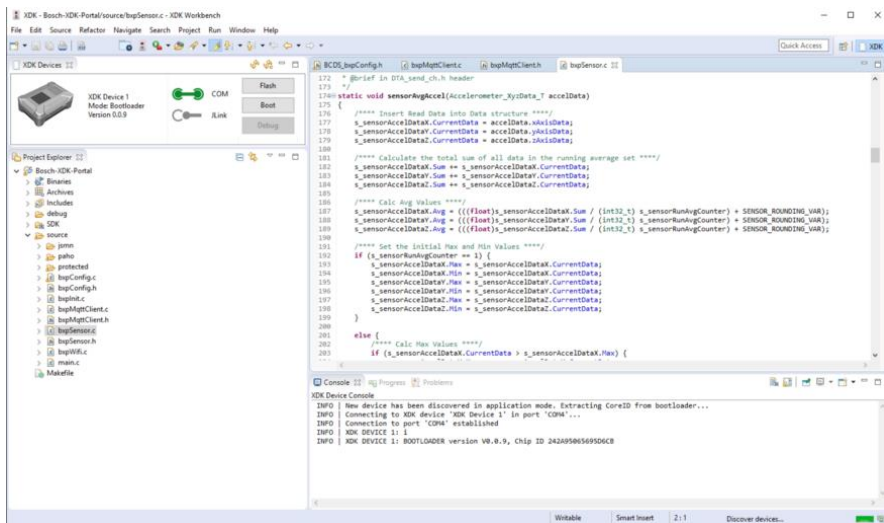


Figure 4

Eclipse Environment of Óbuda University Autonomous Driving Software Sensor C++ system function component to be controlled by Fuzzy-FxLMS system [14]

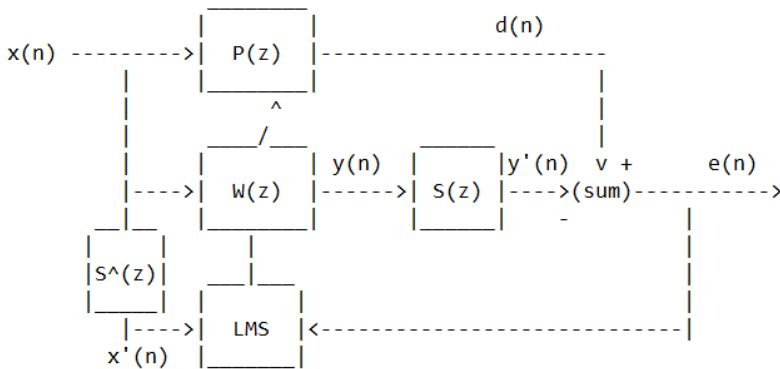
2 F-FxLMS Algorithm for KPI Determination

2.1 FxLMS Algorithm

In this chapter, we are engaged with the classical FxLMS embedded real-time active control of highly autonomous driving software system and controller functions defining embedded performance issues.

The classical (non-fuzzy) FxLMS algorithm can be applied to both feedback and feed-forward structures.

Figure 5, shows the block diagram of a feed-forward FxLMS system.



$x(n)$ is the reference signal
 $P(z)$ is the primary path represented by an IIR filter
 $d(n)$ is the output of $P(z)$
 $w(z)$ is an adaptive filter updated by the FXLMS algorithm
 $y(n)$ is the output of $w(z)$
 $S(z)$ is the secondary path represented by an IIR filter
 $y'(n)$ is the output of $S(z)$
 $e(n) = d(n) - y'(n)$ is the error signal
 $S^{\wedge}(z)$ is a fixed FIR filter from the off-line modeling mode
 $x'(n)$ is a filtered reference signal used by the FXLMS algorithm

Figure 5

“Classical” FxLMS algorithm. In the fuzzy modified version, the off-line modeling mode of the secondary path is realized by a neural network with fuzzy activation functions focused on the convergence optimum of the real-time embedded growing equation

In Figure 5, $P(z)$ resolves the primary path between reference input signal source and error path. $S^{\wedge}(z)$ is purchased offline and fix stored during the real-time transaction of the performance determination, and z is the filter space variable, and n is the time-space variable [12] [13].

The interpretation for the residual error $e(n)$ is given as

$$e(n) = d(n) - y'(n) \tag{1}$$

where:

$y'(n)$ is the filter output $y(n)$ refined through the secondary path $S(z)$,

and $y'(n)$ and $y(n)$ computed as follows:

$$y'(n) = s^T(n)y(n) \tag{2}$$

$$y(n) = w^T(n)x(n) \tag{3}$$

where:

$$w(n) = [w_0(n), w_1(n), \dots, w_{L_i}(n)]^T$$

is a weight vector, and

$$x(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T$$

is the reference signal preferred by the reference signal.

Consequently, there is no feedback from the secondary path to the reference signal.

The FxLMS update equation for the coefficients of $W(z)$ is given as:

$$w(n+1) = w(n) + \mu e(n)x'(n) \quad (4)$$

where $x'(n)$ is the reference signal.

$x(n)$ refined through secondary path model $\hat{S}(z)$, and μ is the gain factor of the adaptive system [6] like:

$$x'(n) = \hat{S}^T(n) x(n) \quad (5)$$

2.2 Fuzzy FxLMS Algorithm

In the adaptive digital signal processing, one of the most usable controllers is the fuzzy-filtered (x) -least mean square (F-FxLMS) algorithm, which is based on the second-order moment of the $e(n)$ error signal (see Fig. 3) with non-periodical reference. The “fuzzy filtered” means that [8] [9]:

The $\hat{S}(z)$ estimation path, the model of the secondary path $S(z)$, is realized with neural nets by fuzzy activation functions. These membership functions could be derived from measurement results so that the real-time growing equation is:

$$w(n+1) = w(n) + \mu_F(n)e(n)x'(n)$$

converges and is optimal. Here the classical gain factor μ is also modified as a function ($\mu_F(n)$) and depends on the error function $e(n)$.

Technically, the new update equation is modified based on the thresholding parameters. In the available autonomous driving systems, these thresholding parameters need to be estimated offline with a secondary path neural network. They cannot be updated during the online operation of autonomous driving detection systems.

For an in-depth study on fuzzy filtered F-FxLMS algorithm with neural nets, the reader may refer to [8] [9].

3 Procedure

The spirit of the embedded KPI algorithm based on the F-FxLMS algorithm can be seen in Figure 6. In this case, the transfer function, realized by a filter $H(z)$ is responsible for the adaptive filtering as a fuzzyfied transfer function between the input vector and the error signal $e(n)$, the “primary path”, while $S(z)$ is the secondary transfer function followed by $W(z)$, which identifies the embedded performance of the software.

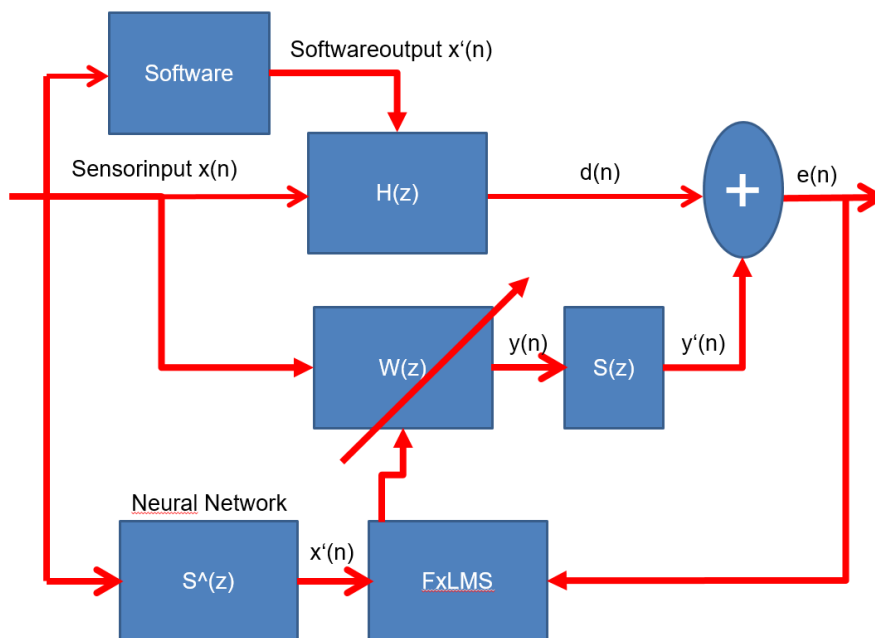


Figure 6

Detailed flow-chart of the procedure

The input video sensor signal $x(n)$ is branched to three paths. First we passed it through the software module to be tested, where the $x'(n)$ is obtained, thru the module algorithm and is passed secondly, to the transfer function $H(z)$.

In the third step, the video signal is applied to the adaptive filter $W(z)$ and finally, to the block expressed by the neural network $S'(z)$, synthesizing the test results, which models the primary path.

We need to compare the output of this with the output of the software module to be tested, which can be done with the coefficients of the adaptive data channel filter $W(z)$.

The core of our F-FxLMS based procedure is that the strength (necessity) of the adaptation is proportional to the key performance of the software module.

That is, the closer we are to the metric of the learned video signals, or the fewer steps we have to realize for convergence, the more efficient the software module.

We can control the comparison and the convergence study with the F-FxLMS algorithm. The task of the primary path ($S(z)$) is to compare the signal $y(n)$ from the adaptive filter and the sensor input and the signal $d(n)$ of the sensor input modified by the software.

Another advantage of our method is that the neural network can be shaped easily by performing a sufficient number of tests. The neural network determines the direction of convergence and the nature of the reference. Otherwise, the method would not be possible in an embedded environment. With the right neural network model, the system can react immediately to changes in the input signal caused by changes in traffic systems. However, the performance of the software system depends primarily on the fuzzified transfer function of the neural network model of the secondary path.

4 Results

The embedded performance algorithm has been approved with the platform software of the Óbuda University (see Figure 2 before) using real data video and radar sequences.

In Figure 7 above, the video image is shown on the left, from which the sensor detects 137 possible objects.

The perception subsystem, where the real-time performance algorithm was placed, detected two objects (red crossed) as automobiles (out of 137), as shown above on the right in Figure 7. The figure below shows an image in a tunnel where the video chip indicated 1209 possible objects, but in reality, there is only one correct object (marked with a red cross).

In the form of crosses in yellow shown in Figure 8, the software error.

This error is expressed in a real-time adaptive manner by the F-FxLMS algorithm.

Figure 9 shows the performance of various measurements. The x-axis shows the time and the vertical axis, the performance value. Square objects are characterized by correct detection to characterize the increase in performance. The excellent quality of the fuzzy adaptive filter is characterized by the fact that the system can respond immediately to changes in the state in milliseconds.

The rectangles are also related to the quality of the video chip. Hardware impulses the appearance of objects.



Figure 7

The video image is shown on the left, from which the sensor detects 137 possible objects. The perception subsystem, where the real-time performance algorithm was placed, detected two objects (yellow crossed) as automobiles (out of 137), as shown above on the right. The figure below shows an image in a tunnel where the video-chip indicated 1209 possible objects, but in reality there is only one correct object (marked with a yellow cross) but now is a wrong position.

Figure 10 shows a comparison of the algorithms (traditional FxLMS and the new fuzzy version we developed) based on 60 endurance (long-range) tests. The average KPI of the test indicated by a dot in the figure. In the figure on the left, where the test using the Video Generation 5 chip shown, it can be observed that the two methods provide approximately the same characteristics. However, there is a significant difference with the use of radar, which is shown in the figure on the right. It can be seen here that the fuzzy algorithm varies much more intensely based on tests than the traditional FxLMS algorithm. This is due to the detection uncertainty of the radar, which can be characterized by the intensification of the uncertainties. This fact is excellently represented by the new F-FxLMS algorithm.

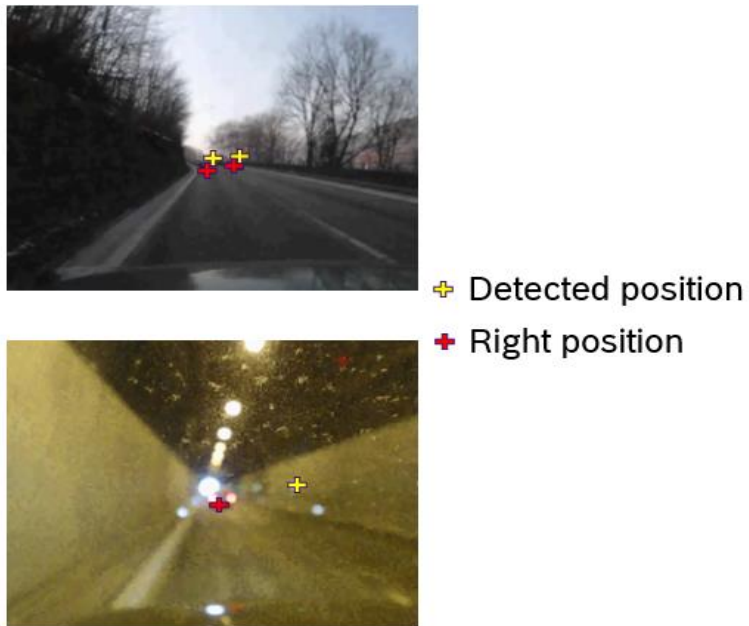


Figure 8

The distance between the detected (yellow) and software-provided (red) centers represents performance, which is determined by the necessary iteration step of the adaptive filter. This error is expressed in a real-time adaptive manner by the F-FxLMS algorithm.

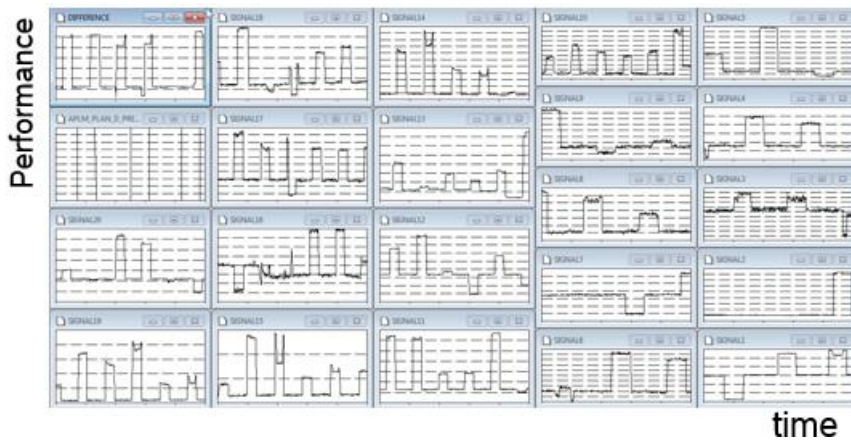


Figure 9

The performance diversity of various HAD video-based measurements. The x-axis shows the time and the vertical axis the performance value. Square objects are characterized by correct detection to represent the increase in performance. The good quality of the adaptive filter is symbolized by the fact that the system is able to respond immediately to changes in the state in milliseconds. The rectangles are also related to the quality of the video-chip

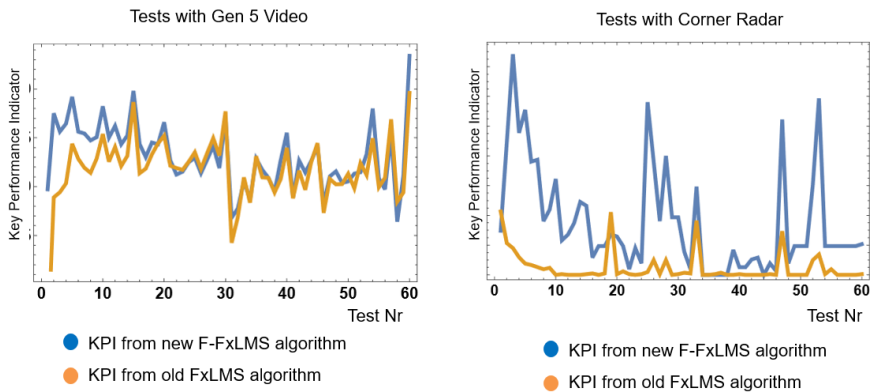


Figure 10

The KPI for video and radar long-range, endurance experiments, for FxLMS and our new fuzzy F-FxLMS algorithm

Conclusions

In this work, we tested the Highly Automated Driving (HAD) project, of Óbuda University. It is important to determine the quality of the various software elements, in an environment characterized by embedded microcontrollers, in real time, in the vehicle.

The advantage of the method, is that by performing a sufficient number of tests, the neural network based digital filter, could be seen as a key object defining the performance of a software module in a real environment. The neural network determines the direction of system convergence and the nature of the reference. Otherwise, the method would not be possible for use in an embedded environment.

With the right neural network model, the system can react immediately to changes in the input signal caused by changes in traffic systems. The performance of the software system depends primarily on the fuzzified transfer function of the neural network model of the secondary path.

During the first part testing the performance indicators of the HAD system, we performed endurance long-range vehicle tests where we can present the performance of the video and radar perception software module as a one-dimensional signal (see Figure 9).

This signal is determined in real time, i.e. when providing control of the vehicle, it provides useful information for possibly switching on other sensors, when performance decreases.

Our future plans for the HAD project and the definition of performance, include:

- 1) Development and implementation of an additional large number of test scenarios;

- 2) Involving the performance of other sensors;
- 3) Examine the performance of the sensor fusion algorithm;
- 4) The systematic alteration of neural networks.

References

- [1] Anderson J. M. et al. 2016: Autonomous vehicle technology. Rand Corporation 2016, p. 214
- [2] Beranek, L. Ver I. L. 1992: Noise and Vibration Control Engineering: Principles and Applications. New York: Wiley, 1992
- [3] Hansen, C. H. Snyder, S. D. 1997: Active Control of Noise and Vibration. London, U. K.: E&FN Spon, 1997
- [4] Szabó, J. Z., and Bakucz, P. 2019: "Determining Algorithmic Performance with Uncertainty Analysis." IEEE 17th International Symposium on Intelligent Systems and Informatics (SISY) IEEE, 2019
- [5] Kuo, S. M., Morgan, D. R. 1996: Active Noise control systems, algorithms and DSP implementation functions, New York, Wiley 1996
- [6] Widrow, M. Stearns, S. D. 1985: Adaptive Signal processing, Prentice Hall, New Jersey 1985
- [7] Kuo, S. M Dipa Vijayan, M. D. 1997: A Secondary path modeling technique for Active Noise Control Systems. IEEE Transactions on Speech and Audio Processing, July 1997
- [8] Zadeh, L. A. Fuzzy Sets. Information & Control. 1965, Vol. 8, pp. 338-353, ISSN 0019-9958
- [9] Cintula, P. From Fuzzy Logic to Fuzzy Mathematics. PhD Thesis. FJFI CVUT Praha, 2004
- [10] Mathworks Inc.: Matlab 2018b computer algebra software. System Identification Toolbox
- [11] EE Automotive 5th generation of video sensors offers significantly higher resolution <https://www.eenewsautomotive.com/news/>
- [12] Obeidi, T et al, 2018: Fuzzy logic-based maximum power point tracking for a solar electric vehicle. Acta Polytechnica Hungarica, Vol. 15, No. 7, 2018
- [13] Y. Sun et al, 2019: A fuzzy brain emotional learning classifier design and application in medical diagnosis. Acta Polytechnica Hungarica, Vol. 16, No. 4, 2019
- [14] Óbuda University: HAD Software Users Manual. Highly Autonomous Driving Platform. Budapest 2019