The Redundant Virtual Sensors via Edge Computing

Peter Peniak
University of Žilina,
Faculty of Electrical Engineering and Information Technology, Department of Control and Information Systems
Žilina, Slovak Republic
peter.peniak@feit.uniza.sk

Emília Bubeníková
University of Žilina,
Faculty of Electrical Engineering and Information Technology, Department of Control and Information Systems
Žilina, Slovak Republic
emilia.bubenikova@feit.uniza.sk

Alžbeta Kanáliková
University of Žilina,
Faculty of Electrical Engineering and Information Technology, Department of Control and Information Systems
Žilina, Slovak Republic
alzbeta.kanalikova@feit.uniza.sk

Abstract – In the paper, we share our research results and experiences with the redundant virtual sensors. The main focus is paid on the creation of the model for redundant virtual sensors, applying the concept of "digital twins" that are associated with physical sensors. Virtual sensor can be implemented as a software instance in Cloud (SaS) or physical device via Edge computing. The goal is to create a common model which would enable a generic approach independently on vendors' implementation. MQTT protocol was selected as the main integration platform that is supported by various IoT devices and mainly sensors. To enable feasible implementation of a redundant sensor, Edge computing is used as a platform to host digital twins for physical devices. The redundant sensors can be applied for variety fail-safe applications, such control systems of tunnels.

Keywords – IoT, MQTT, Virtual sensors, Digital twins, Edge

I. Introduction

The fail-safe applications require reliable and validated inputs from physical sensors in order to steer critical processes applying high availability and fail-safe methods. The physical sensors are the sensors that produce a signal based on stimuli from the external environment, e. g. they record the state of the air, temperature, radioactivity, etc. The physical sensor transmits a resulting impulse – typically through electrical signals that can be captured and stored in digital form [1].

The essential idea of the presented proposal is to enrich measured results of the physical sensors and calculate new and validated results that would be considered a fail-safe. This means that it is necessary to safety the system for integrity against faults and accidental errors [2]. To achieve this goal, the concept of digital twins (DT) can be applied.

The classic concept of the digital twins means creating a virtual model of a process, product or service to accurately reflects a physical object. The components are connected to a cloud-based system that receives and processes all the data that the sensors monitor. This pairing of the virtual and physical worlds allows for example data analysis and optimization of the production process. In our case, digital twins could be associated to a group of selected sensors, enriching values of physical sensors with additional fail-safe algorithms according to the requested level of redundancy. This approach can simplify the development and operations of process control application with no need to perform additional fail-safe operations, that are embedded in the digital twin, acting for control applications as the virtual sensor.

Then several identical sensors can be used to sense each quantity and one virtual sensor is assigned to this "group of sensors". This means that redundant sensors are considered in the process, which increase the reliability of the measurement. The output value of the virtual sensor, which is further provided to the fail-safe application, is obtained based on of a calculation that depends on the required degree of reliability. In our paper, we consider 2 scenarios: a comparison of values from 2 sensors and comparison from 3 sensors (2 from 3).

The main goal of the article is the creation of the model for redundant virtual sensors, applying the concept of "digital twins" that are associated with physical sensors, as mentioned before. These virtual sensors could be implemented as fail-safe sensors without dependency on the features of specific sensors of various providers. Digital twins in form of virtual sensor have to have access to physical sensors and enrich their values according to selected redundancy model. Virtual sensor can be implemented as a software instance in Cloud (SaS) or physical device via Edge computing. The integration of physical sensors with associated virtual sensor would be achieved via Internet of Thing protocols, such as MQTT.

This integration creates a possibility to be as close as to the control process. This proposal aims to support critical and real-time business and control systems while using cloud services. In this paper, we share our research results and experiences with the modeling and implementation of redundant virtual sensors.

II. Virtual sensors

A virtual sensor is a software form of a sensor that also processes measurements of several physical sensors [3]. The virtual sensors can open access to information about the low level of the physical sensors, they improve the precision of the physical sensors or creates physical sensor interoperability.

The most common reasons for using virtual sensors are the following:

- information cannot be obtained or calculated using a physical device,
- the physical sensors used are very limited due to their properties and capabilities,
- there is a high probability that the sensor results may be incorrect,
- the sensors work in demanding conditions with a possible failure.

Figure 1 shows one of the possible implementations of a group of virtual sensors based on physical sensors. Sensor data is stored in virtual clouds [3].

ISBN 978-80-261-0973-0, © University of West Bohemia, 2021

2021 International Conference on Applied Electronics (AE) | 978-80-261-0973-0/21/$31.00 ©2021 10.23919/AE51540.2021.9542888

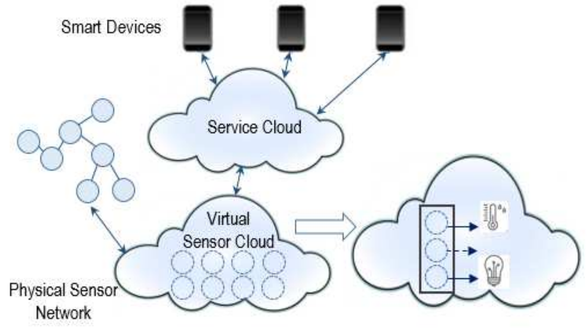


Fig. 1. Virtual sensors [3]

The connection of virtual and physical sensors (PS) can have different constellations, which is shown in Fig. 2. For example: two physical sensors and one virtual sensor (VS) (a - example), one physical sensor and multiple virtual sensors (b - example), multiple physical sensors, and multiple virtual sensors each (c - example) [4].

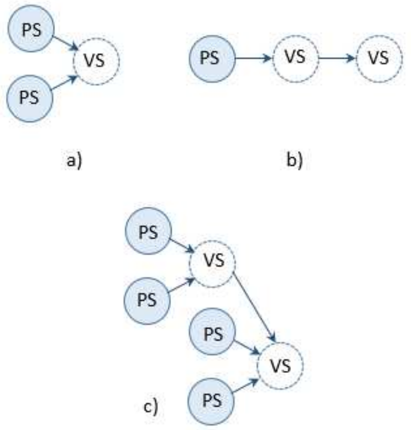


Fig. 2. Virtual sensor connection with physical sensors [4].

The virtual sensor provides enriched information, which it obtains through mathematical models to support control and other processes. The mathematical models used to obtain enriched information are based, for example, on methods [5]:

- Fuzzy logic, this approach e. g. applies simple rules of fuzzy logic when fusing values from multiple physical sensors and finding connections between them.
- Self-organizing neural networks, for example for prediction, for troubleshooting, e. g. for the classification of electrical faults in induction motors, etc.
- Other methods, for example, SVM - Support Vector Machine, probabilistic latent variable models, deep learning, etc.

Estimators of process and product data, for example, use of Kalman filters, or swarm optimization algorithm, ant colony optimization or genetic algorithm, which have flexibility in application in various tasks, one of the best known is the optimization method, etc. [5], [6], [7].

The derived measurements through mathematical models represent the virtual sensor data. To storage this data to persist this data, a digital twin is required. Digital twin combines and integrates active data sources and controls their availability and validity [4].

Virtual sensors represent a software layer that provides indirect measurements of a process variable, or an abstract condition based on data gathered by physical (or other virtual) sensors leveraging a fusion function [8], [9].

A digital twin is a software that takes real data about a physical system and estimates how that physical object or system will be affected by those inputs [10], [11], [12]. The digital twin is meant to be an up-to-date and accurate copy of the physical object's properties and states, including shape, position, gesture, status and motion and other values [13].

A digital twin may be used for simulation, as a kind of prototype to understand expected behavior, and can also capture real-world behavior so that, for example, analytics and learning can be performed.

In order to clearly describe the concept of a virtual sensor and also to identify key properties, Fig. 3 graphically illustrates its building blocks and their relationships.

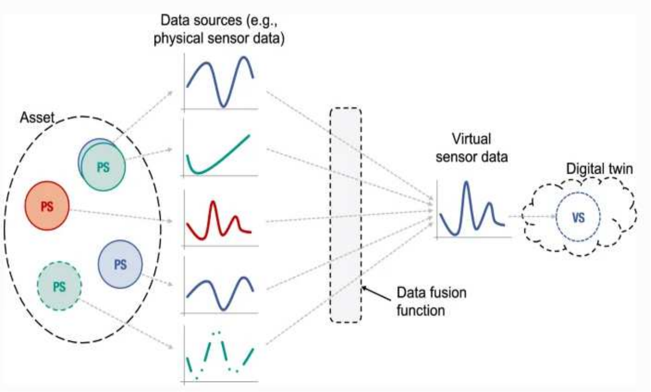


Fig. 3. Virtual sensor concept [4].

III. The Model for Virtual Redundant Sensors

According to the proposed concept, the model of virtual redundant sensor is based on digital twins. The physical sensors are mapped to their digital twins and provide the real measured values by sensors. The fusion function F_i (Comparator "*m* from *n*") is used to transform measured values to the validated value of the virtual sensor. Enriched validated value is then provided by the virtual redundant sensor to business-critical applications, as shown in Fig. 4.

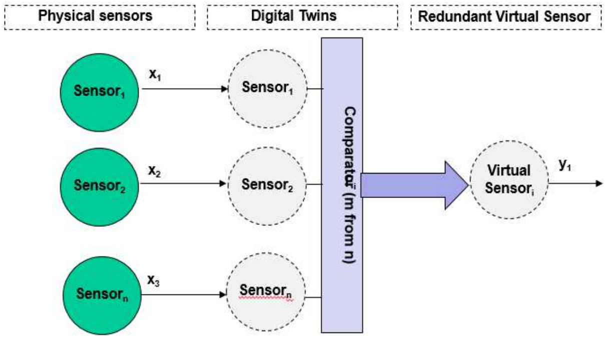


Fig. 4. Virtual redundant sensor – model

The numerical model of the proposed virtual redundant sensor can be represented by formulas (1) to (11). The general fusion function is described by formula (1), where result y is derived from M-sensors out of all N sensors (x_i).

$$y = F_{MN}(x_1, x_2,..., x_n) \qquad (1)$$

The numerical model of virtual sensors has two designed typical scenarios with functions F_{22} and F_{23}. The basic fusion function F_{22} is designed for the comparison of the two sensors, as expressed by formula (2). The result y is derived according to the difference between sensors values. If the difference exceeds the allowed threshold k, the result is set to “0”, which means the sensor cannot provide the valid value (3). If the threshold is not reached, the result is calculated as the arithmetic average of sensors values (4).

$$y = F_{22}(x_1, x_2) \qquad (2)$$

$$y =0, \qquad | x_1 - x_2 | > k \qquad (3)$$

$$y= (x_1 + x_2)/2, | x_1 - x_2 | < k \qquad (4)$$

Another typical use-case would be the fusion function F_{23} for comparison 2 from 3 sensors, as expressed by formula (5). The result is derived as an arithmetic average of sensors in pair, with the minimum difference value (6-11).

$$y = F_{23}(x_1, x_2, x_3) \qquad (5)$$

$$z_{12}=| x_1 - x_2 | \qquad (6)$$

$$z_{13}=| x_1 - x_3 | \qquad (7)$$

$$z_{23}=| x_2 - x_3 | \qquad (8)$$

$$y= (x_1 + x_2)/2, z_{12}=Min\{z_{12}, z_{13}, z_{23}\}< k \qquad (9)$$

$$y= (x_1 + x_3)/2, z_{13}=Min \{z_{12}, z_{13}, z_{23}\} <k \qquad (10)$$

$$y= (x_2 + x_3)/2, z_{23}=Min\{z_{12}, z_{13}, z_{23}\}<k \qquad (11)$$

$$y = 0, \qquad z_{12}, z_{13}, z_{23} > k \qquad (12)$$

where

F_{MN}	- fusion function for M from N physical sensors
y	- calculated value of virtual redundant sensor (VS$_i$)
x_i	- measured value of physical sensor (Sensor$_i$)
z_{ij}	- difference of physical sensor values (S$_{i,j}$)
k	- defined threshold of comparator (max. difference)

IV. IMPLEMENTATION OF VIRTUAL REDUNDANT SENSORS

For the implementation of virtual redundant sensors, Edge computing approach is selected. Edge computing is defined as a distributed architecture with decentralized processing power. In edge computing, sensor data is processed by the local computer instead of its transferring for processing to Cloud, or remote data center. The main reason for Edge computing is to support requirements with processing in real time and very low latency.

Edge device works as an additional hardware box (Proxy node) for the associated physical sensors, see Fig. 5. The results from physical sensors are available for Edge device only. It means that sensors’ data cannot be directly reachable for external systems. Instead, Edge device provides a validated value of the virtual redundant sensor, as outcome of the calculation of physical values. The generated value is validated, therefore can be used by an external system of applications as reliable value without necessity for another application processing and insuring of fail-safe operations.

Edge device has to have an embedded MQTT broker, that is used for its communication with MQTT physical sensors (Sensor_S$_i$). MQTT protocol is based on the “Publish/Subscribe” model with the central “Broker instance”, which creates the concept of a “global data space” that is accessible to all interested applications. All communication is represented as reads and writes to the global data space [14], [15]. The physical sensors are connected to Edge’s MQTT broker and can publish their measured physical results to associated MQTT topics (Topic_S$_i$), according to formula (12). The measured values are read from MQTT Broker by Edge kernel software via MQTT client. Kernel software transforms values of physical sensors via defined fusion function F_{22}, or F_{23} to the validated enriched value, which is provided as the new result of the virtual redundant sensor (Sensor_VS$_1$) for external systems, as expressed by formula (13). In addition, Edge device can publish this new result also into its embedded MQTT broker via Topic_VS$_1$.

$$Topic\_S_i = Sensor\_S_i \qquad (13)$$

$$Topic\_VS_1= Sensor\_VS_1 \qquad (14)$$

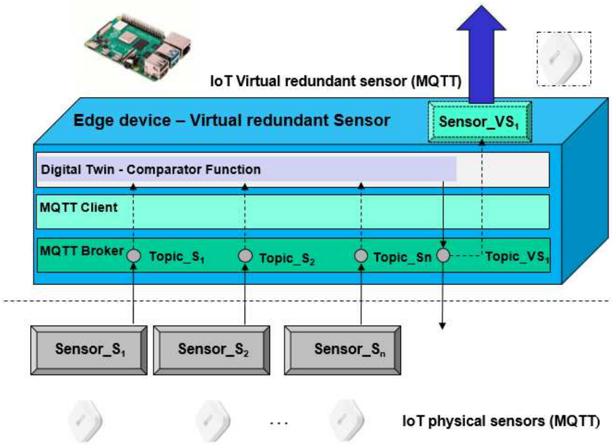

IoT Virtual redundant sensor (MQTT)
Edge device – Virtual redundant Sensor
Sensor_VS1
Digital Twin - Comparator Function
MQTT Client
MQTT Broker
Topic_S1
Topic_S2
Topic_Sn
Topic_VS1
Sensor_S1
Sensor_S2
Sensor_Sn
...
IoT physical sensors (MQTT)

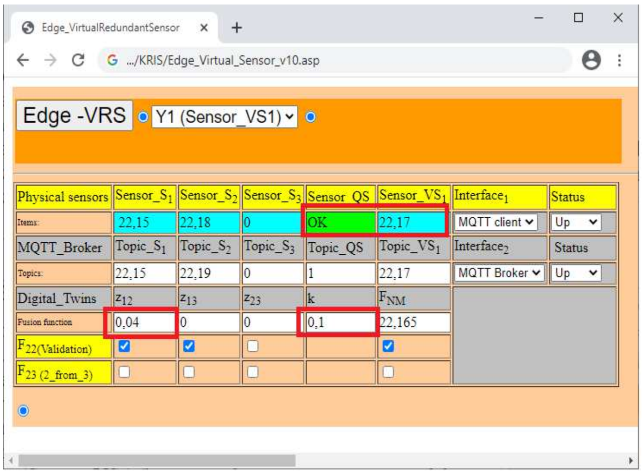
Fig. 5. Implementation of virtual redundant sensor via Edge device.

V. VALIDATION OF PROPOSED EDGE MODEL

The virtual redundant sensor concept was validated in our laboratory. We have used a virtual machine for emulation of Edge device with Virtual Redundant Sensor application (Edge-VRS) including MQTT Broker (Rabbit MQ) and WEB server (IIS, ASP.NET), for configuration and internal management of Edge-VRS application. The functional validation was performed with both fusion functions F_{22} and F_{23}.

The testing of fusion function F_{22} is documented in Fig. 6 and Fig. 7. The physical values from Sensor_S$_1$ and

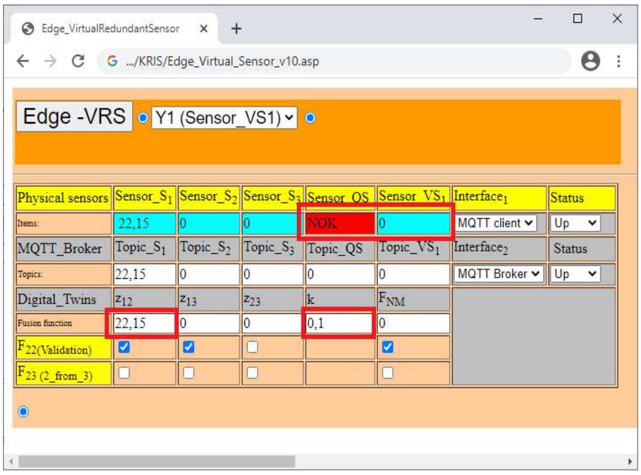
Sensor_S$_2$ are published to MQTT broker and the fusion function calculates their deviation. If deviation z_{12}=0,04 is not exceeding the defined threshold value (k=0,1), the virtual sensor provides a validated result (OK), which is indicated by MQTT topic Sensor_QS. In addition, the virtual sensor provides transformed value 22,17 to MQTT topic Topic_VS$_1$ of associated virtual sensor Sensor_VS$_1$, according to formula (4).


Edge_VirtualRedundantSensor
.../KRIS/Edge_Virtual_Sensor_v10.asp
Edge -VRS
Y1 (Sensor_VS1)

Physical sensors	Sensor_S$_1$	Sensor_S$_2$	Sensor_S$_3$	Sensor_QS	Sensor_VS$_1$	Interface$_1$	Status
Items:	22,15	22,18	0	OK	22,17	MQTT client	Up
MQTT_Broker	Topic_S$_1$	Topic_S$_2$	Topic_S$_3$	Topic_QS	Topic_VS$_1$	Interface$_2$	Status
Topics:	22,15	22,19	0	1	22,17	MQTT Broker	Up
Digital_Twins	z_{12}	z_{13}	z_{23}	k	F_{NM}		
Fusion function	0,04	0	0	0,1	22,165		
$F_{22 (Validation)}$	☑	☑	☐		☑		
$F_{23 \ (2\_from\_3)}$	☐	☐	☐		☐		

Fig. 6. Virtual redundant sensor with fusion function F_{22} -validation OK.

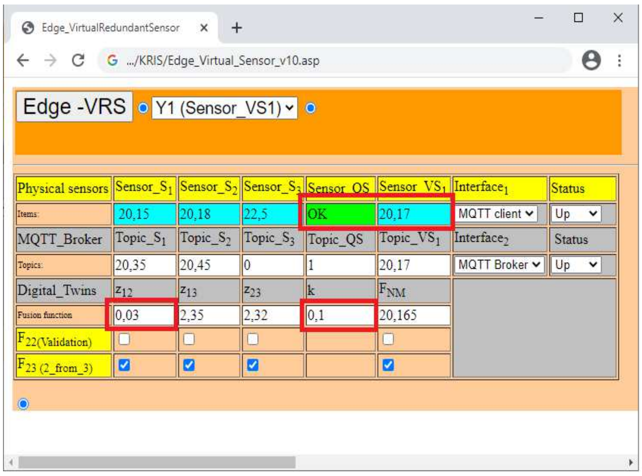
In case if the deviation z_{12} exceeds the defined threshold (k = 0,1), then the virtual sensor cannot provide a validated result (NOK) and topic Topic_VS$_1$ has to have the default value 0, according to formula 3. This case is emulated and shown in Fig. 7, with disconnected physical sensor Sensor_S$_2$.


Edge_VirtualRedundantSensor
.../KRIS/Edge_Virtual_Sensor_v10.asp
Edge -VRS
Y1 (Sensor_VS1)

Physical sensors	Sensor_S$_1$	Sensor_S$_2$	Sensor_S$_3$	Sensor_QS	Sensor_VS$_1$	Interface$_1$	Status
Items:	22,15	0	0	NOK	0	MQTT client	Up
MQTT_Broker	Topic_S$_1$	Topic_S$_2$	Topic_S$_3$	Topic_QS	Topic_VS$_1$	Interface$_2$	Status
Topics:	22,15	0	0	0	0	MQTT Broker	Up
Digital_Twins	z_{12}	z_{13}	z_{23}	k	F_{NM}		
Fusion function	22,15	0	0	0,1	0		
$F_{22 (Validation)}$	☑	☑	☐		☑		
$F_{23 \ (2\_from\_3)}$	☐	☐	☐		☐		

Fig. 7. Virtual redundant sensor with fusion function F_{22} -validation NOK.

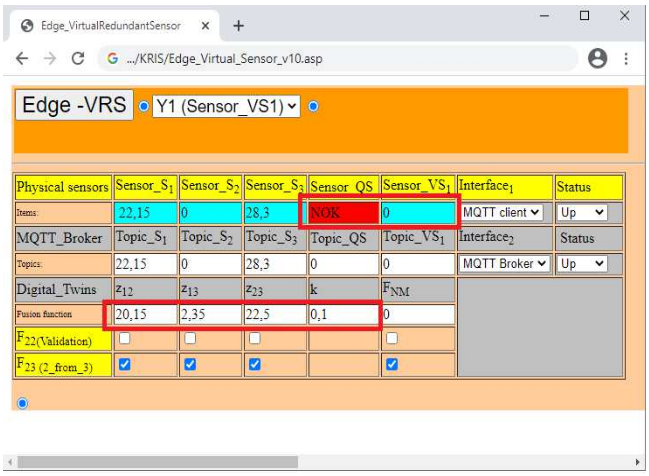
In addition, we have tested the extended validation model with fusion function F_{23}. There were 3 physical sensors connected with MQTT broker. The fusion function calculated differences between all sensors (z_{12}, z_{13}, z_{23}) and used sensors with minimal value difference (Sensor_S$_1$, Sensor_S$_2$), where sensor Sensor_S$_3$ reported the wrong result. The value of virtual sensors was again calculated by formula (9) and stored to Topic_VS$_1$, as shown in Fig. 8.


Edge_VirtualRedundantSensor
.../KRIS/Edge_Virtual_Sensor_v10.asp
Edge -VRS
Y1 (Sensor_VS1)

Physical sensors	Sensor_S$_1$	Sensor_S$_2$	Sensor_S$_3$	Sensor_QS	Sensor_VS$_1$	Interface$_1$	Status
Items:	20,15	20,18	22,5	OK	20,17	MQTT client	Up
MQTT_Broker	Topic_S$_1$	Topic_S$_2$	Topic_S$_3$	Topic_QS	Topic_VS$_1$	Interface$_2$	Status
Topics:	20,35	20,45	0	1	20,17	MQTT Broker	Up
Digital_Twins	z_{12}	z_{13}	z_{23}	k	F_{NM}		
Fusion function	0,03	2,35	2,32	0,1	20,165		
$F_{22 (Validation)}$	☐	☐	☐		☐		
$F_{23 \ (2\_from\_3)}$	☑	☑	☑		☑		

Fig. 8. Virtual redundant sensor with fusion function F23 -validation OK.

The similar test was performed with one disconnected sensor Sensor_S$_2$, in this case, the result was still validated in contrast to fusion function F_{22} from the previous testing. That to simulate not validated result, we even had to let sensor Sensor_S$_3$ publish a significantly different value, where difference exceeded defined threshold. The absence of value from one sensor and threshold bust finally caused that result could not be validated as illustrated in Fig. 9.


Edge_VirtualRedundantSensor
.../KRIS/Edge_Virtual_Sensor_v10.asp
Edge -VRS
Y1 (Sensor_VS1)

Physical sensors	Sensor_S$_1$	Sensor_S$_2$	Sensor_S$_3$	Sensor_QS	Sensor_VS$_1$	Interface$_1$	Status
Items:	22,15	0	28,3	NOK	0	MQTT client	Up
MQTT_Broker	Topic_S$_1$	Topic_S$_2$	Topic_S$_3$	Topic_QS	Topic_VS$_1$	Interface$_2$	Status
Topics:	22,15	0	28,3	0	0	MQTT Broker	Up
Digital_Twins	z_{12}	z_{13}	z_{23}	k	F_{NM}		
Fusion function	20,15	2,35	22,5	0,1	0		
$F_{22 (Validation)}$	☐	☐	☐		☐		
$F_{23 \ (2\_from\_3)}$	☑	☑	☑		☑		

Fig. 9. Virtual redundant sensor with fusion function F_{23} -validation NOK.

VI. Conclusion

In this paper, we share our research results and experiences with the modelling and implementation of redundant virtual sensors. The essential idea was based on the approach to construct virtual sensors, which would be able to validate the results and enable application independence on vendors' implementations. The critical applications would not need to deal with the fail-safe and redundant operations, instead use validated and reliable results directly from Edge devices covering sensors even various vendors.

The sensor integration platform was based on MQTT (IoT) protocol, with an embedded MQTT broker on the Edge device, that is broadly supported by vendors of IoT devices and sensors. The used digital twin approach included the fusion functions F_{MN} for both selected numerical models (2 sensors, 2 from 3 sensors). In fact, depending on the number of redundant sensors, the virtual sensor model was capable to validate and calculate the correct result for the critical applications, even with one disconnected sensor. The results of testing have shown that the proposed virtual redundant sensor can be used for validation of results based on redundant sensors via Edge device approach and digital twin. Although the result of testing confirmed the proposed model, any further extension of the model could help to cover additional use-cases for critical applications. As well, instead of using the virtual machine concept for Edge device, the physical implementation based on selected hardware would provide a more precise evaluation of the proposed model closer to real technical implementation.

ACKNOWLEDGMENT

Support of the project APVV 17–0014, “Smart tunnel: telematics support in case of emergencies in the road tunnel”.

REFERENCES

[1] Fraden, J.: Handbook of Modern Sensors, Physiscs, design, and Applications, Springer International Publishing, 2016, 978-3-319-19302-1, 10.1007/978-3-319-19303-8

[2] Rástočný K.: Quantitative Assessment of the Diagnostics Effect on the Hardware Safety Integrity of the Safety-Related Electronic System Operating in Low Demand Mode of Operation, RELIABILITY AND SAFETY OF ELECTRONICS DEVICES, 2019, Volume 17, number 2.

[3] Virtual Sensor Working and Its Applications, [online] https://www.elprocus.com/virtual-sensor-working-applications/

[4] Martin, D., Kühl, N. and Satzger, G. Virtual Sensors. Bus Inf Syst Eng (2021), [online] https://doi.org/10.1007/s12599-021-00689-w/

[5] Xianglin Zhu, Khalil Ur Rehman, Bo Wang and Muhammad Shahzad: Modern Soft-Sensing Modeling Methods forFermentation Processes, https://www.mdpi.com/1424-8220/20/6/1771/pdf

[6] Bhagya R. Navada, Santhosh K. Venkata and Swetha Rao, A Soft Sensor for Estimation of In-Flow Rate in a Flow Process Using Pole Placement and Kalman Filter Methods, journal Machines,Volume 7, Issue 4 2019, [online] https://www.mdpi.com/2075-1702/7/4/63

[7] Gregor, M. , Spalek, J.: Fitness-based Adaptive Control of Parameters in Genetic Programming, 2016 preprint, [online] https://arxiv.org/ftp/arxiv/papers/1605/1605.01514.pdf

[8] Nemec, D., Janota, A. Hruboš, M. and Šimák,V.: Intelligent real-time MEMS sensor fusion and calibration, IEEE Sensors J., vol. 16, no. 9, pp. 7150-7160, Oct. 2016.

[9] NY Lii, S. Sturm a TA Coombs, Redundant Dissimilar Sensor Fusion with Dynamic Driver Input Classification and Graceful Degradation for Drive-by-Wire Applications,“ 2010 IEEE 71. Vehicular Technology Conference , Taipei, Taiwan, 2010, pp. 1-5

[10] Gartner’s Top 10 strategic Technology Trends for 2017, [online] https://www.gartner.com/smarterwithgartner/gartners-top-10-technology-trends-2017/.

[11] Kaur, M. J., Mishra, V. P., & Maheshwari, P. (2020). The convergence of digital twin, IoT, and machine learning: transforming data into action. In Digital twin technologies and smart cities (pp. 3-17). Springer

[12] Alam KM, El Saddik A (2017) C2PS: a digital twin architecture reference model for the cloud-based cyber-physical systems. IEEE Access 5:2050–2062.

[13] Negri E, Fumagalli L, Macchi M (2017) A review of the roles of digital twin in CPS-based production systems. Procedia Manuf 11:939–948].

[14] ISO/IEC 20922:2016 [ISO/IEC 20922:2016, Information technology - Message Queuing Telemetry Transport (MQTT) v3.1.1 (online) https://www.iso.org/standard/69466.html

[15] Peniak, P. – Bubeníková, E. – Kanáliková, A.: Extended gateway model for OPC UA/IoT device integration, In: SAMI 2021 - IEEE 19th World Symposium on Applied Machine Intelligence and Informatics, ProceedingsPages 155 - 15921 January 2021