

Article

Simulation of Virtual Redundant Sensor Models for Safety-Related Applications

Peter Peniak ^{*} , Karol Rástočný, Alžbeta Kanáliková  and Emília Bubeníková 

Department of Control and Information Systems, Faculty of Electrical Engineering and Information Technology, University of Žilina, 01026 Žilina, Slovakia; karol.rastocny@feit.uniza.sk (K.R.); alzbeta.kanalikova@feit.uniza.sk (A.K.); emilia.bubenikova@feit.uniza.sk (E.B.)

* Correspondence: peter.peniak@feit.uniza.sk

Abstract: Applications of safety-related control systems demand reliable and credible inputs from physical sensors, therefore there is a need to extend their capabilities to provide a validated input with high availability. Our main idea is to insert virtual sensors between physical sensors and the control system's logic. The created solution can validate the values of real sensors and with the use of multiple virtual sensors we can achieve high availability in addition, therefore our solution is entitled as a virtual redundant sensor. It works by the digital twin's concept and uses fusion function to calculate validated results. The fusion function is used to transform the measured values from the physical sensors according to designed numerical models. The selection of a numerical model with assigned fusion functions can be performed via the WEB-based graphical user interface. Proposal of the numerical model is created and validated on the experimental workplace with emulation of physical sensors and MQTT integration (smart IoT sensors). The results of testing have shown that our solution can be applied to validate the values of physical sensors. Proposed fusion functions calculated results according to the selected model in all cases, while non-standard cases were handled according to our definition. In addition, the high availability concept with a group of two virtual sensors has proven fast recovery and availability of results for safety-related applications as well.

Keywords: virtual redundant sensor; fail-safe; safety-related application; fusion function; digital twin; MQTT; simulation; software tool RabbitMQ; safety; availability



Citation: Peniak, P.; Rástočný, K.; Kanáliková, A.; Bubeníková, E. Simulation of Virtual Redundant Sensor Models for Safety-Related Applications. *Sensors* **2022**, *22*, 778. <https://doi.org/10.3390/s22030778>

Academic Editors: Hossam A. Gabbar and Carosena Meola

Received: 16 November 2021

Accepted: 17 January 2022

Published: 20 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

There are processes (e.g., in industry and transport) that are associated with the occurrence of a dangerous (hazardous) event that can result in damage to human health and the environment, or significant material damage. Such undesirable events or their consequences may be prevented by the application of technical and organizational security measures. Technical safety measures also include safety-related control systems (SRCs). Sensors are also a part of such systems.

The safety-related applications in process control require credible and validated inputs from physical sensors (PSs) to control safety-related processes using methods that guarantee not only the required level of safety level but also high availability. Therefore, our article will deal with virtual redundant sensors.

Virtual sensor (VS) is, by one of the available definitions, a digital twin of the physical device, or logical/digital representation of a real physical sensor, whose capabilities are enriched. Although the virtual sensor is seen as a real sensor, it exists just logically or is provided by the IT infrastructure (Cloud/Edge). The virtual sensor can even exist when the physical sensor is not present. It is a similar approach as virtualization of computers (virtual machines), networks (VLANs), and software components.

The virtual redundant sensor is our new term defined by us in [1] because we want to suggest the use of the virtual sensors not only as a representant of the physical sensors with

validation of physical sensors' values but also to achieve their high availability by applying for a redundancy. The classical approach assumes that each application includes the necessary features to support the validation of sensors values, as shown by Figure 1a. This approach naturally leads to the higher complexity of applications and requires repetition of the same programming steps with each application. The solution according to Figure 1a assumes redundancy only at the level of physical sensors.

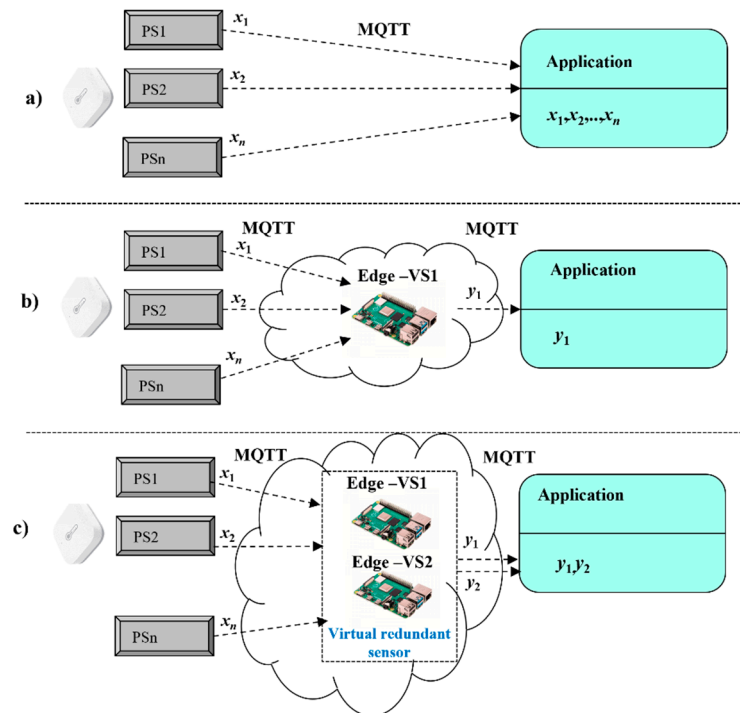


Figure 1. Different models for safety-related applications and sensors (a) classical model, (b) original model and (c) extended model.

This article extends our initial work [1], where we developed the original concept for virtual redundant sensor and created its basic numerical model and validated solution based on a dedicated Edge device with an MQTT broker. In order to simplify the design of safety-related applications, we introduced the dedicated Edge device for processing all sensors' values with the calculation of validated results, based on a selected numerical model (*moon*), as shown by Figure 1b. The solution according to Figure 1b contains redundancy not only at the level of physical sensors but also software redundancy applied inside the virtual sensor.

Although we have been able to validate physical sensor values, the created solution has been vulnerable in case of Edge failure. In such a case, the application would not be able to work with sensors' values and would be inoperable. Therefore, there was a need to revise our original concept and elaborate on the target solution. First, our essential idea is to extend the original concept of the virtual redundant sensor with flexible calculation models according to the required use cases, without the necessity to create the new solutions, case by case. Second, fail-safe conditions require a solution with high availability, so we are proposing the new model, shown in Figure 1c. It is based on the group of two dedicated Edge devices, which are mirrored via AMQP protocol, and could take-over services each other to support applications in fail-safe approach. The solution according to Figure 1c contains redundancy not only at the level of physical sensors but also at the level of virtual sensors.

2. Virtual Sensors

The concept of the Internet of Things and the Industrial Internet of Things addresses connectivity and communication anytime, anywhere, for anything and anyone [2]. On the one hand, it facilitates data acquisition, on the other hand, failures in these systems can cause accidents and undermine credibility in the eyes of the public. Therefore, it is necessary to focus on the means of solving reliability such as fault avoidance, fault tolerance, fault rectification [3,4].

The physical sensors can measure values with errors, and this can lead to the instability of other systems that need data from the measuring sensors and can lead to an accident. Today, we already know systems for detection, and fault isolation diagnostics (FID). The aim of FID systems is to reach a system that is resistant to failures (failure, erroneous measurements), which will retain the functionality of the existing system or systems that receive data from that system. Such a system control approach is called the fault-tolerant control (FTC) approach [5,6]. Sensors are increasingly found in a wide range of applications, such as intelligent mobile devices, traffic management, automotive systems, industry [7–9], healthcare, environmental monitoring, health monitoring, and the like.

The virtual sensor is a software emulation of the physical sensor which retrieves values from basic physical sensors. Virtual sensors work often in a Cloud system (Software as a Service) and can work in multiple configurations in conjunction with physical sensors: one PS connected to one VS (one-to-one), one PS connected to multiple VSs (one-to-many), multiple PSs connected to one VS (many-to-one), multiple PSs connected to multiple VSs (many-to-many) [10], in Figure 2.

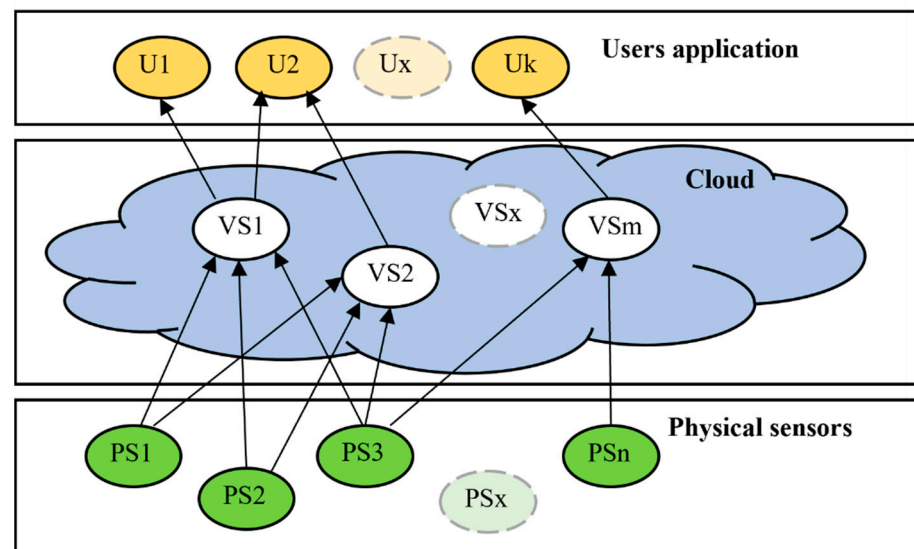


Figure 2. Virtual sensors connection.

The development of virtual sensors has significantly helped in the control processes of industrial, transport, and other processes, especially when seen from the point of view of accuracy and availability. In the literature review, virtual sensors are defined as [11]:

- Mathematical models for predicting process behavior;
- Mathematical and statistical models to enrich the physical sensors values;
- A combination of new analytics methods to generate new value in real-time.

The same review [11] of the literature and the studies lists the criteria that were used in the implementation of virtual sensors:

- In controlled process monitoring–fault detection, detail fault, estimation, performance degradation;
- In process optimization–optimization parameters, scheduling, supply chain.

In [12–17], it is possible to find some interesting applications of virtual sensors in areas such as:

- Traffic—vehicle suspension control by means of virtual sensors based on a neural network to determine the estimate vehicle unsprung mass relative velocity [12];
- Environment—virtual sensing for real-time monitoring of surface and groundwater sources for irrigation purposes [13];
- Health—use of a virtual sensor for safety monitoring of a person’s position as a replacement for a real position sensor, provided that the position matches within the active radius of the sensor [14];
- Working environment—temperature monitoring in the indoor environment (offices, living spaces, industrial environment) and prediction of temperature development using genetic programming [15];
- Industry—A virtual sensor for detecting and resolving a human physical collision with conventional industrial robots using two torque observers [16];
- Agriculture—implementation of virtual sensors designed to monitor the temperature in greenhouses [17].

In [18–20], it is possible to find applications for the use of virtual sensors to increase the functional safety or diagnosis of virtual sensor failures, such as:

- Maintaining the functional safety state of direct current converters (DC-DC) converters using hardware-physical sensors and techniques based on advanced extended Kalman filter (EKF) and virtual sensors [18];
- Fault detection and diagnosis sensor of the chiller in air conditioning systems using virtual sensors constructed by a long-short-term memory network [19];
- Fault detection and diagnostics in ventilation units using virtual sensors designed to measure deviations of values [20] and many others.

Based on the above information, it is appropriate to implement virtual sensor applications in terms of:

- Optimization, prediction, or data analysis;
- Safety, availability, and fault detection.

3. The Numerical Model of the Virtual Sensor for Safety-Related Applications

According to the proposed concept, the virtual sensor model is based on digital twins. Physical sensors are assigned to their digital twins and provide the actual values that are measured by the sensors. The fusion function (1) is used to transform the measured values (values from the physical sensors) to the value provided by the virtual redundant sensor. The value thus obtained from the virtual sensor is then provided to safety-related applications, as shown in Figure 3.

In general, the function of fusing signals from sensors can be described (1), where y is the resulting signal from the virtual redundant sensor, x_i is the signal from the i -th physical sensor, and n is the number of the physical sensors.

$$y = F(x_1, x_2, \dots, x_n). \quad (1)$$

The aim of SRCs (which in a broader sense that can include the concept of a virtual redundant sensor) is to have the required availability and safety features. These properties are expressed by the so-called RAMS (Reliability, Availability, Maintainability, Safety) parameters. The relationship between these parameters is shown in Figure 4. The concept of a virtual redundant sensor does not preclude its use in applications where high availability and minimum safety requirements are required.

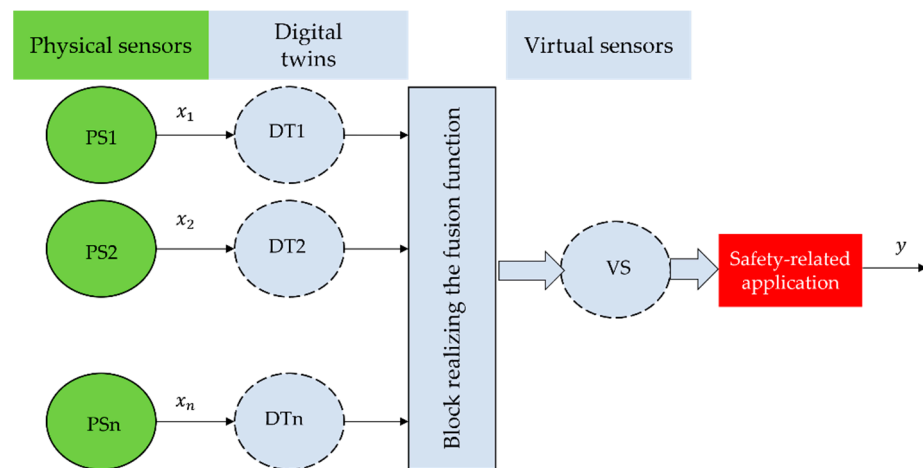


Figure 3. Virtual sensor model for safety-related application.

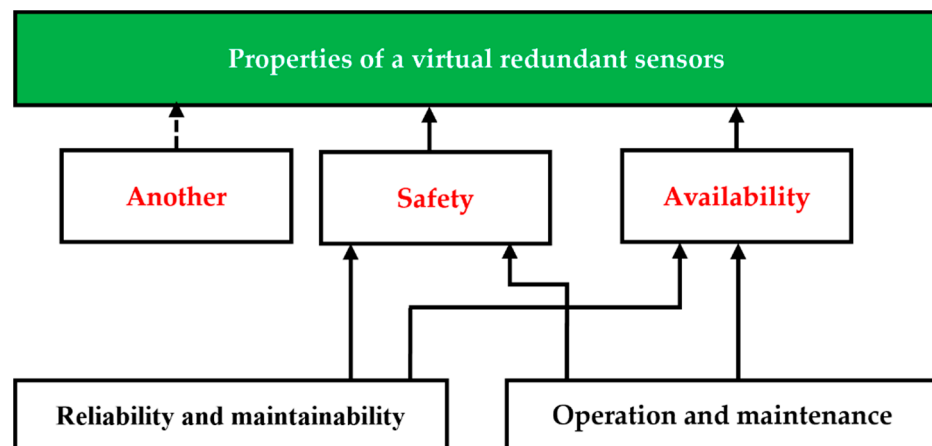


Figure 4. The relationship between the RAMS parameters and virtual redundant sensors.

The amount of hardware and software redundancy depends on requirements for the availability and the safety integrity of such a virtual redundant sensor. In this article, we consider the processing of values in the virtual redundant sensor from a maximum of three physical sensors. The prerequisite for using the virtual redundant sensor is that the signals from the physical sensors correspond to the same physical quantity.

Depending on the application in which the virtual redundant sensor is to be used, signal processing from n physical sensors can be realized in order to:

- Increase the safety of values at the output of the virtual redundant sensor (mathematical model n out of n , in the following text $noon$, where $n > 1$);
- Increase the safety and availability of values at the output of the virtual redundant sensor (mathematical model m out of n , in the following text $mooon$, where $1 < m \leq n$).

Physical sensor redundancy can only be used to increase the safety of information from physical sensors (model $noon$, if $n = 2$, then model is $2oo2$). Then the function (1) has the form:

$$y_{2oo2}(t) = F_{2oo2}(x_1(t), x_2(t)). \quad (2)$$

The requirement of safety demands the definition of:

- Tolerable range (TR) of difference of values from physical sensors;
- A method of selecting a signal value at the output of the virtual redundant sensor if the physical sensors provide signals of different values; in general, depending on the application, either the smallest value or the largest value can be considered safer; in the case of a logic signal, it is usually a signal with a log. level 0 [21–23]; in the next

part of the article, we assumed that the safest value is considered to be the smallest value (3) of the physical sensor signals (a similar consideration can be made if the safest value would be considered the largest value of the physical sensor signals);

- The safe state of the virtual redundant sensor (output signal value) if the signal values from the physical sensors are out of the tolerable range; in this case, the virtual redundant sensor will leave this safe state only as a result of corrective intervention by an authorized worker.

Example of using the $\min\{\}$ function: Let a dangerous state be considered to be a state when the liquid level in the tank is below a defined critical value. The liquid level is measured by two physical sensors. A safe response (opening of the liquid filling valve) is activated when at least one of the physical sensors reports that a critical value has been reached.

Example of using the $\max\{\}$ function: Let a dangerous state be considered to be a state when the liquid level in the tank is above a defined critical value. The liquid level is measured by two physical sensors. The safe response (opening of the liquid filling valve) is activated when at least one of the physical sensors reports that a critical value has been reached.

Then

$$\begin{aligned} \text{if } |x_1(t) - x_2(t)| \leq TR, \text{ then } y_{2oo2}(t) &= \min\{x_1(t), x_2(t)\}, \\ \text{if } |x_1(t) - x_2(t)| > TR, \text{ then } y_{2oo2} &= 0. \end{aligned} \quad (3)$$

If it is a logic signal, the resulting value from the virtual redundant sensor can be determined based on the relationship:

$$y_{2oo2} = x_1 \cdot x_2. \quad (4)$$

If the disagreement of the logic signal levels at the output of the physical sensors ($x_1 \neq x_2$) lasts longer than the defined tolerable disagreement time t_{TD} , then $y_{2oo2} = 0$ and this value can change only due to the corrective intervention of the authorized worker.

Theoretically, it is also possible to consider the *noon* model, where $n > 2$, but such a solution comes into consideration only in a very specific situation.

If the redundancy of physical sensors is used in order to increase the safety and availability of values from physical sensors (it is about model *moon*, if $m = 2$ and $n = 3$), then it is the model 2oo3. Then the function (1) has the form:

$$y_{2oo3}(t) = F_{2oo3}(x_1(t), x_2(t), x_3(t)). \quad (5)$$

If it is an analog signal and it applies:

$$\begin{aligned} |x_1(t) - x_2(t)| &= k_1 \leq TR, \\ |x_1(t) - x_3(t)| &= k_2 \leq TR, \\ |x_2(t) - x_3(t)| &= k_3 \leq TR, \end{aligned} \quad (6)$$

then

$$y_{2oo3}(t) = \min\{x_1(t), x_2(t), x_3(t)\}. \quad (7)$$

If that is true

$$(k_1 \leq TR) \wedge (k_2 > TR) \wedge (k_3 > TR), \quad (8)$$

then

$$y_{2oo3}(t) = \min\{x_1(t), x_2(t)\} \quad (9)$$

and the physical sensor PS3 (Figure 3) is logically isolated from the virtual redundant sensor ($x_3(t) = 0$) and the virtual redundant sensor operates in the degraded mode (model 2oo2) until taking the corrective intervention of the authorized worker. Subsequently, based on this intervention, the virtual redundant sensor goes into full mode (model 2oo3).

If that is true

$$(k_1 > TR) \wedge (k_2 \leq TR) \wedge (k_3 > TR), \quad (10)$$

then

$$y_{2003}(t) = \min\{x_1(t), x_3(t)\} \quad (11)$$

and the physical sensor PS2 (Figure 3) is logically isolated from the virtual redundant sensor ($x_2(t) = 0$) and the virtual redundant sensor operates in the degraded mode (model 2002) until taking the corrective intervention of the authorized worker as a result. Subsequently, based on this intervention, the virtual redundant sensor goes into the full mode (model 2003).

If that is true

$$(k_1 > TR) \wedge (k_2 > TR) \wedge (k_3 \leq TR), \quad (12)$$

then

$$y_{2003}(t) = \min\{x_2(t), x_3(t)\} \quad (13)$$

and the physical sensor PS1 (Figure 3) is logically isolated from the virtual redundant sensor ($x_1(t) = 0$) and the virtual redundant sensor operates in the degraded mode (model 2002) until taking the corrective intervention of the authorized worker. Subsequently, based on this intervention, the virtual redundant sensor goes into full mode (model 2003).

In all other combinations of signal values from physical sensors, the output $y_{2003}(t) = 0$ until corrective intervention is taken by an authorized worker. In the case of a logic signal, the resulting value from the virtual redundant sensor can be determined by voting based on the relationship:

$$y_{2003} = x_1 \cdot x_2 + x_1 \cdot x_3 + x_2 \cdot x_3. \quad (14)$$

If the disagreement of the logic signal levels at the output of the physical sensors ($x_1 \neq x_2$, or $x_1 \neq x_3$, or $x_2 \neq x_3$) lasts longer than the defined tolerable disagreement time t_{TD} and $y_{2002} = 0$, the same procedure is followed as in the case of analog signal if a non-functional physical sensor can be identified, it is disconnected and the virtual redundant sensor goes from the full mode (model 2003) to the degraded mode (model 2002).

Theoretically, it is also possible to consider the *moon* model, where $m > 2$ and $n > 3$, but such a solution comes into consideration only in a very specific situation.

Calculations and processing of values must take place in an environment characterized by a high safety level that corresponds to the safety level of the signal provided by the virtual redundant sensor.

In Section 3 of this article, we provide only a basic mathematical description of the problem. The algorithms actually used are significantly more complex due to the minimization of hazardous (undesirable) states in collision situations due to the change of signals at the outputs of physical sensors.

4. Extended Model of the Virtual Redundant Sensor

As explained in Section 3, the proposed model of the virtual sensor is primarily designed to provide the validation of physical sensors' values with the selectable fusion function and mathematical model according to Formulas (1) till (14). Yet, there is a risk of virtual sensor failure, which would lead to the overall malfunction of the connected application. Our virtual sensor had to be extended with respect to redundancy, which would ensure its availability. There are several options how to achieve the requested redundancy. This article deals with two possible approaches on how to implement redundancy. Our extended model consists of two identical virtual sensors that are implemented on dedicated independent physical Edge devices, performing validation of physical sensors' values by using the same fusion Function (1). If one of them fails, the other must step in and support the application with sensors' values.

In this article, the focus was paid on two alternative approaches:

1. Virtual redundant sensors with mirroring;
2. Virtual redundant sensors with duplexing.

Virtual redundant sensors with mirroring was our first approach (Figure 5). Let us assume that one of the virtual sensors is always active VS1P (virtual primary sensor), while the second one VS1B (virtual backup sensor) is a backup. VS1B just mirrors all physical sensors' values from VS1P via replication process. AMQP connection is used for replication of sensors values (x_1, x_2, \dots, x_n) , which are mirrored to associated AMQP queues. VS1B can hold values even if the connection with VS1P is broken. VS1P works in active mode and communicates with physical sensors and applications via MQTT and performs validation of sensors values according to the chosen model (*moon*). The validated results are provided by VS1P (15). If VS1P fails, VS1B will step in to support application with sensors' values.

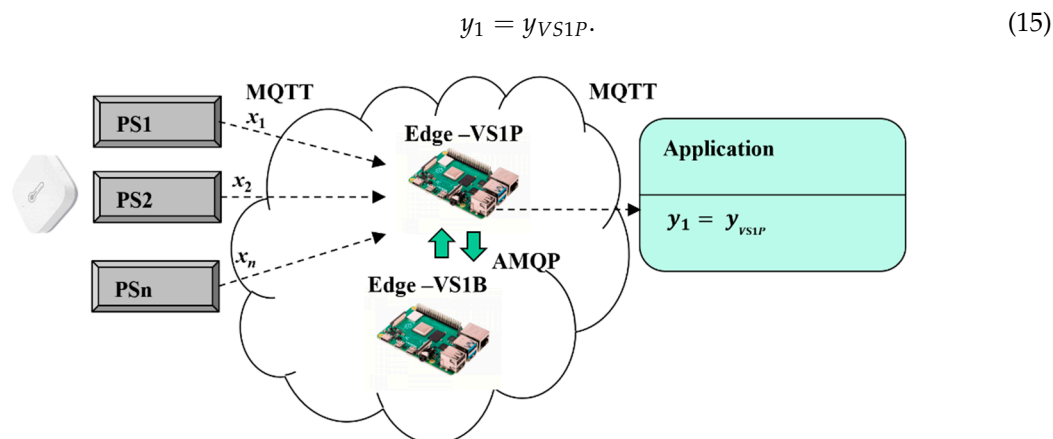


Figure 5. Virtual redundant sensor model with mirroring.

Figure 6a illustrates the operations of the virtual redundant sensor in case of VS1P failure. After VS1P is broken (Event 1) all communications channels are closed. VS1B recognizes the unavailability of VS1P and takes over the role of the active virtual sensor including his IP address. Temporarily unavailable virtual sensor VS1P will be switched by common configuration to the backup role in case of a restart to prevent looping of connections (Event 2). After reconnection of application (Event 3) and physical sensors (Event 4), the validation process (*moon*) can be restarted again based on mirrored physical sensors values in AMQP queues, as shown by Figure 6b.

$$y_1 = y_{VS1B}. \quad (16)$$

Although the virtual redundant sensor model with mirroring enables fast recovery according to fail-safe principles, there is a still need for some transition time T_T to recover operation after time when VS1P had failed. All the physical sensors and applications have to be configured with a forced MQTT reconnection process to MQTT broker in case the connection is lost. As soon as a failure of VS1P is recognized by VS1B, a recovery process is initiated. VS1B takes-over IP address of VS1P and the forced reconnection process starts by physical sensors and applications. The overall reconnection time of physical sensors T_{PSR} can be expressed by (17), as a maximum of reconnection times of all physical sensors $T_{PS1R}, T_{PS2R}, \dots, T_{PSnR}$. Then overall transition time T_T can be calculated as maximum from reconnection time of application T_{AR} and reconnection time of physical sensors T_{PSR} according to (18). After transition time is reached, the virtual redundant sensor is recovered and results are provided by VS1B. Formula (19) covers safety-related requirement of applications.

$$T_{PSR} = \max\{T_{PS1R}; T_{PS2R}; \dots; T_{PSnR}\}, \quad (17)$$

$$T_T = \max\{T_{PSR}; T_{AR}\}, \quad (18)$$

$$T_T \leq T_{ST}, \quad (19)$$

where T_{ST} is the tolerable time interval during which application can work without validated virtual sensors' values (according to an application specific safety-related requirements).

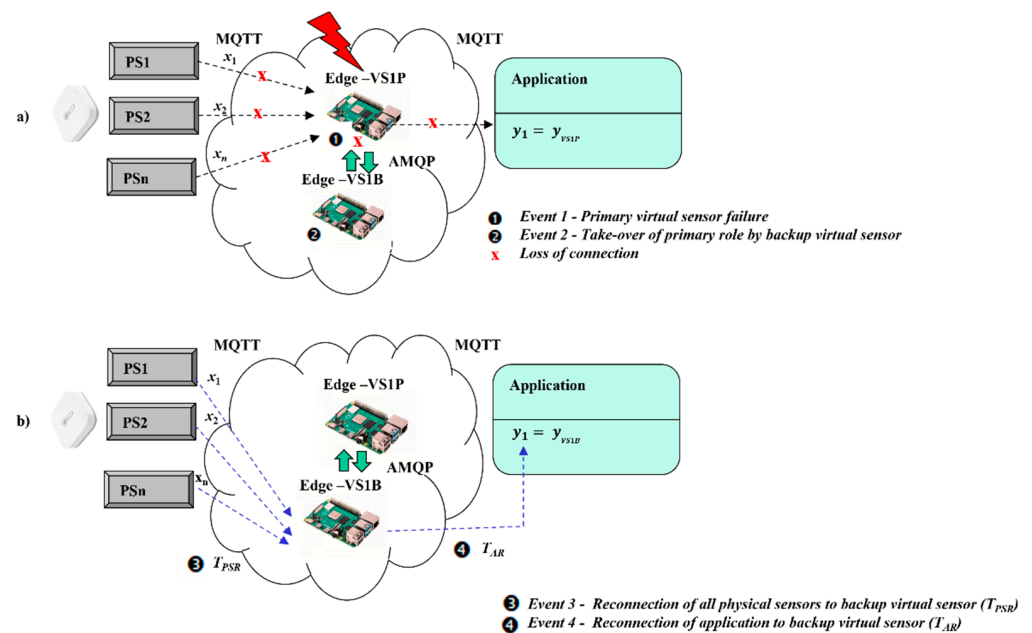


Figure 6. Virtual redundant sensor model with mirroring in case of VS1P failure.

To overcome the mentioned imperfections of the mirroring approach, an alternative virtual sensor model with duplexing was suggested. The main goal is to mitigate the temporary unavailability of virtual redundant sensors and overall transition time ($T_T = 0$) so that safety-related requirements of application could be kept. There are again two virtual sensors used (VS11, VS12), but in contrast to the previous model, they are both active modes having connections with physical sensors and applications. It means that the validation of physical sensor values (x_1, x_2, \dots, x_n) is performed simultaneously with the use of selected fusion functions (*moon*) by two virtual sensors on Edge devices. There are several aspects to be taken into consideration:

- An application has to be able to receive two validation results (y_{VS11}, y_{VS12});
- A possible delay with the delivery of the results has to be handled properly (T_{VS11}, T_{VS12});
- A pairing of the results has to be insured by the setting of the result identifier (R_{id});
- The virtual sensors have to communicate together and align the results with parameter (p); if $p = 0$ both virtual sensors can provide results, if $p = 1$ the result is valid only from one virtual sensor.

The principle of the duplexing approach is explained in Figure 7. Physical sensor values are received by both sensors (Event 1). Alignment of the results requires the arbiter function, which is performed by the leading sensor. Arbiter function is pre-selected on VS11, but it can be taken-over by VS12 in case of VS11 failure. As soon as the leading virtual sensor calculates the result, the incremental identifier R_{id} is assigned and sent with input parameters to VS12 (Event 2). VS12 assigns for the result the same R_{id} and confirms it to VS11 via AMQP (Event 3). VS11 lets result parameter without change ($p = 0$), as validated by both sensors. In case of communication with VS12 is not possible, VS11 sets $p = 1$ indicating that results from VS12 to be ignored (possible malfunction). Both virtual sensors can send their results together (Event 4) with R_{id}, p in slightly different times based on network conditions (T_{VS11}, T_{VS12}), according to Formulas (20) and (21). In the case of parameter $p = 0$, the application can use any of the results. In the case of parameter $p = 1$, application has to use only associated result from the same virtual sensor.

$$y_1 = \{y_{VS11}, R_{id}, T_{VS11}, p\}, \quad (20)$$

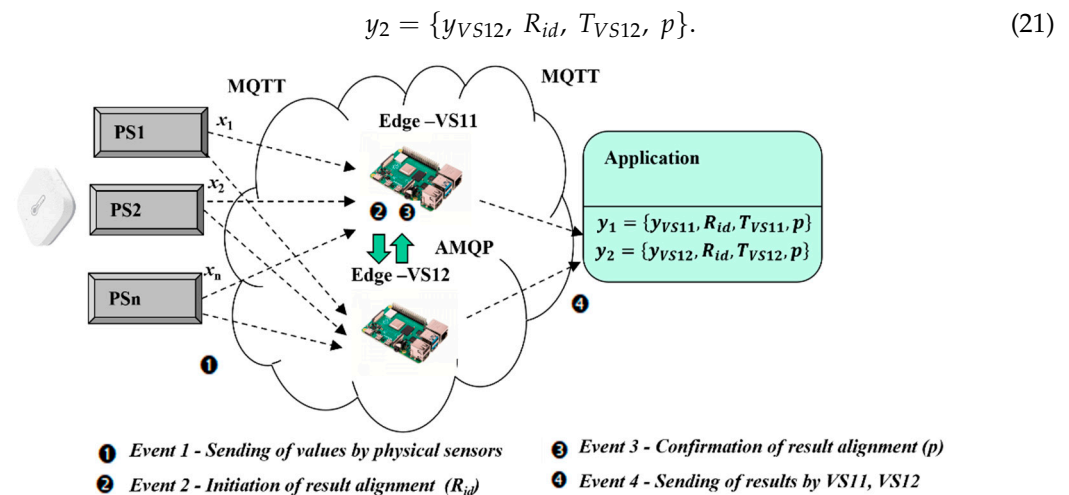


Figure 7. Virtual redundant sensor model with duplexing approach.

5. Experimental Workplace for Simulation of Virtual Redundant Sensors

The proposed numerical model of the virtual redundant sensor, described in Section 4, underwent experimental validation in our laboratory. First, we designed experimental solution to test the various numerical models of virtual redundant sensors having taken into consideration the possible implementation scenarios. The following aspects have been considered during the design phase of our experimental solution:

- Physical sensors have to support the most common IoT protocols;
- Integration layer for sensors has to be broadly supported and known;
- Virtual redundant sensor shall support various implementation scenarios;
- Mathematical models shall be configurable and with the modular approach.

Based on these preconditions, we have chosen MQTT as an integration layer for physical sensors, acting as IoT devices. MQTT works on top of the TCP/IP (Transmission Control Protocol/Internet Protocol) protocol stack as a lightweight broker-based publish/subscribe messaging protocol. MQTT protocol is based on “Publish/Subscribe” model with central “Broker instance”, which creates the concept of a “global data space” that is accessible to all interested applications [24–26]. All communication is represented as reads and writes to the global data space. This chapter has aligned all symbols between software screens and text so that it can be better understood. As well, we use symbol VS as a simplified name for virtual redundant sensors in our next text.

The physical sensors are connected to MQTT broker as a publisher, writing measured values (x_i) into associated Topics_ x_i . Virtual redundant sensor, on the contrary, is connected as subscriber, reading all values from Topics_ x_i . Afterwards, the values of physical sensors can be evaluated according to the selected numerical model (F_{moon}) and transformed according to associated formulas (F_{moon}) to the new value of virtual redundant sensor VS_i (y_1), and the virtual redundant sensor can support multiple calculations in the same time (for example y_1 means 1st calculation).

In addition to the subscriber role, the virtual redundant sensor has to have another connection to the broker, in the role of Publisher, so that it can publish its result to MQTT broker. The new value is written to the associated Topic_ y of virtual redundant sensor (Figure 8). The control applications can access values of virtual redundant sensors in the same way as values of physical ones, via MQTT broker and associated Topic, in order to abstract implementation and virtual nature of the new sensor (digital twin to physical sensors).

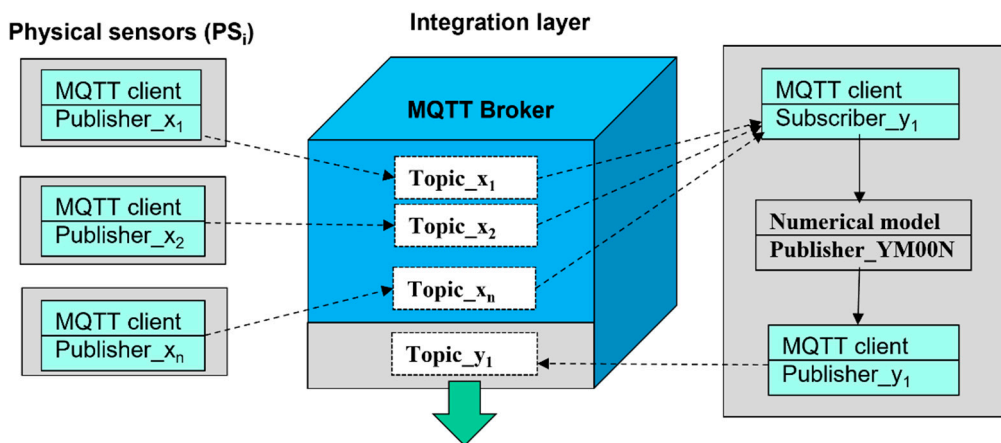


Figure 8. The concept of experimental workplace for simulation of virtual redundant sensors.

As MQTT broker we chose Rabbit MQ software [27,28], which is mainly used as AMQP (Advanced Message Queuing Protocol) broker to support advanced messaging and queue management with high availability and performance. However, it supports additional protocols (MQTT) and features in the form of additional “plug-ins”.

To implement overall experimental solution, the following parts were used:

- ASP.NET script for emulated physical, virtual redundant sensors;
- Rabbit MQ as MQTT broker, as described before;
- MQTT clients for producers, consumers.

RabbitMQ (software management tool Rabbit MQ) management console is shown in Figure 9, with configured connections of the physical sensors (blue box, IP address: 192.168.111.2 . . . n) in publisher roles and the virtual redundant sensor in both publisher and consumer roles (192.168.111.1).

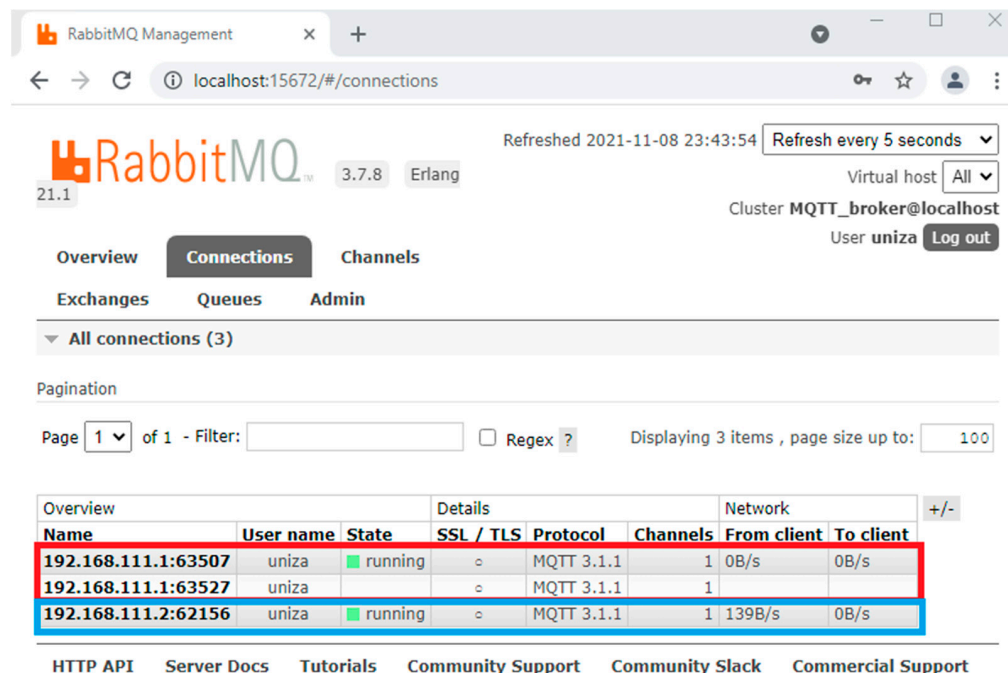


Figure 9. RabbitMQ as MQTT broker with sensor connections.

The consumer role requires proper setting of the AMQP (Advanced Message Queuing Protocol) queue (*amq.topic*) with routing keys that are mapped to MQTT topic of the sensors

($topic_{x1}, topic_{x2}, \dots, topic_{x_n}$) for IP address 192.168.111.1 of virtual redundant sensor, as shown in Figure 10.



Figure 10. RabbitMQ queue setting for virtual redundant sensor as a consumer.

In our experimental case, the physical sensors are emulated purely by software client, which can send manual input, or simulated values to MQTT broker via MQTT client in Producer role. A simulation of the input is based on the random value generator, which calculates deviation level for initial value, that is configurable on the screen of the emulated physical sensor (GUI).

Our implemented experimental workplace is shown in Figure 11. The emulated physical sensors generate measured values every 10 s to MQTT client, which sends them to MQTT broker with associated routing key “topic_x1”, that is assigned to MQTT queue (*amq.topic*) on the broker side and represents associated topic of the physical sensor PS1. The virtual redundant sensor is connected to MQTT broker via MQTT client in consumer role, reading values of connected physical sensors.

The simulations of proposed virtual redundant sensor were performed on the implemented experimental workplaces. The goal was to evaluate all numerical models (*moon*) with the emulated input from three physical sensors PS1–PS3. All sensors were represented by MQTT clients which were producing emulated measured values with updating of values as soon as the value has been changed in contrast to previous value.

Figure 12 shows GUI of all three emulated sensors with the actual emulated values, which were sent to MQTT broker from emulated physical sensors as was illustrated by Figure 11.

- PS1: $x_1 = 22.79$;
- PS2: $x_2 = 22.40$;
- PS3: $x_1 = 21.20$.

The most essential part of our work was dedicated to the creation of virtual redundant sensor software with support of defined numerical models in Equations (1) to (14). In order to enable a flexible simulation, the software client of the virtual redundant sensor supports selection of defined models with its required parameters. Virtual redundant sensor’s user interface (GUI—Graphical User Interface) is displayed in Figure 13.

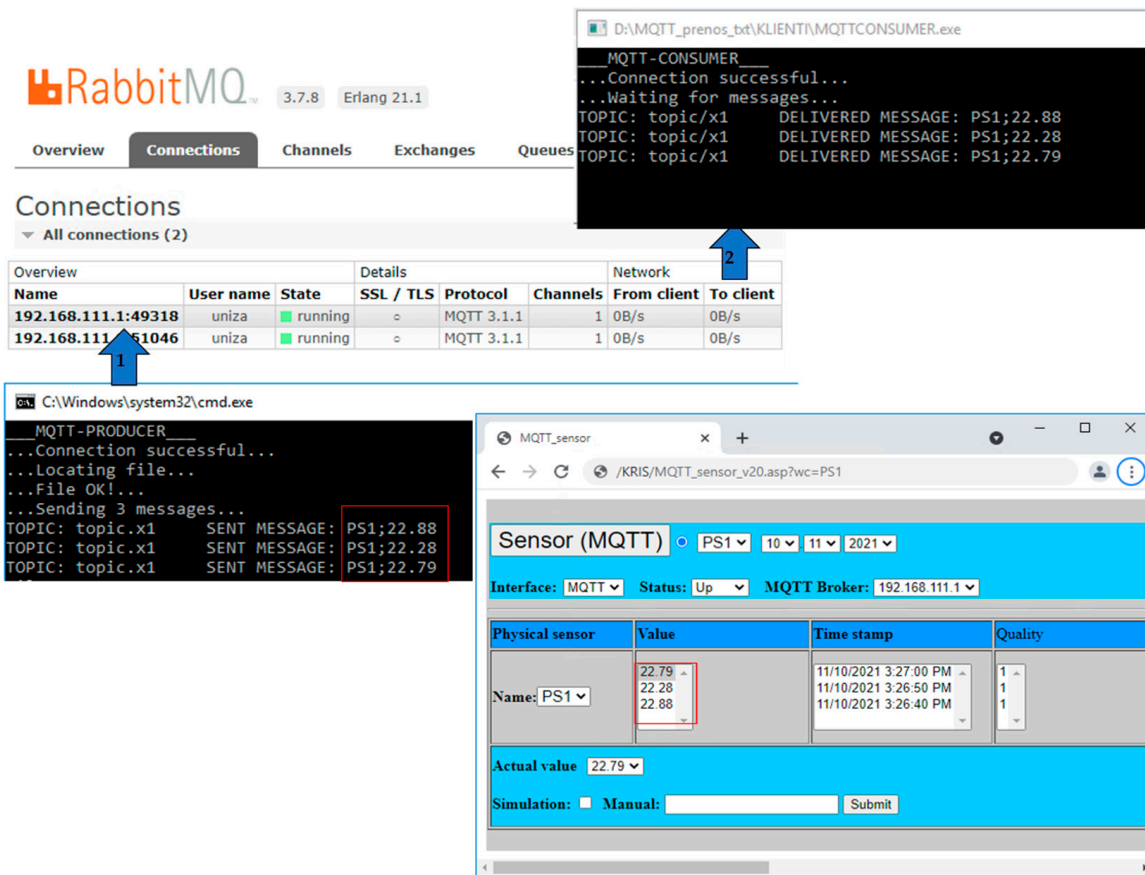


Figure 11. Emulated physical sensor with MQTT client.

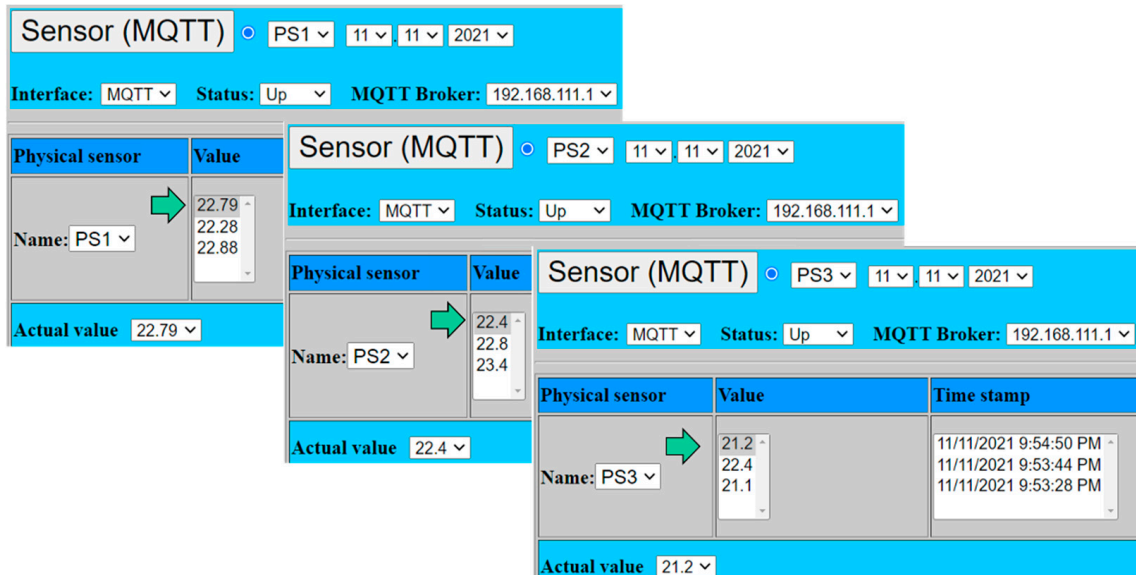


Figure 12. Emulated physical sensors connected to virtual sensor.

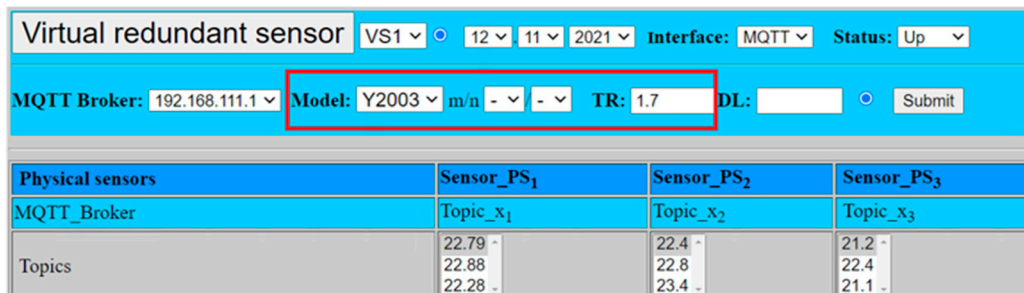


Figure 13. Emulated physical sensors connected to virtual sensor.

GUI shows all actual values of connected physical sensors (PS_i), which are mapped to associated topics on MQTT broker ($Topic_{x_i}$), including the history of the last recent values (2 values back). The selection of required numerical models ($moon$) including the required parameters is marked by red box.

For confirmation of the selected model, the submit button has to be activated. Software will automatically pre-select the desired model, confirming the selection of the model and its parameters in informational part of GUI: YM00N, TR .

There is a need to select the following parameters:

- n —number of physical sensors ($n \geq 1$ and $n \leq 3$);
- m —number of evaluated sensors ($m \leq n$, $n \geq 1$ and $n \leq 3$);
- TR —tolerance range (float number $x.y$);

As soon as the model is selected, in our case Y2003, software calculates the result value of the virtual sensor VS_i . The calculated value is automatically transferred into associated topic on the MQTT broker ($Topic_{y_1}$), which is to be consumed by fail-safe applications requiring the value's validation. In addition, GUI provides quality status (validity) and time stamp so that evaluation of the results can be performed properly. In this most complex case, the values of all three sensors are to be evaluated and included into the overall calculated result.

Figure 14 illustrates one of the possible scenarios, when parameter $TR = 1.7$ and tolerable differences between all three sensors are met (0.39, 1.2, 1.59). Therefore, validation result is positive and can be set to OK. The result value is calculated as a minimum value out of all three sensor values ($VS1 = 21.2$), see equation (22). This model would support positive validation result even in case of higher deviation from one of the sensors, the calculated result would be based on values from the remaining two sensors that met TR level, setting result value to the minimum of two sensors. The negative validation result would be calculated only if two sensors show deviation higher than the pre-defined TR value.

$$VS1 = y_{2003} = F_{2003}(x_1, x_2, x_3) = \min\{22.79, 22.40, 21.20\} = 21.20 \quad (22)$$

In order to demonstrate the negative validation result, parameter TR was set to 1.0. In this case, the values from sensors PS_1 , PS_2 show smaller deviation (0.39) than defined limit TR (1.0), however the differences k_2 , k_3 between the values of the sensors PS_2 , PS_3 (1.2) and PS_1 , PS_3 (1.59) exceed the defined limit (1.0). Therefore, according to (7), (9), and (11), defined conditions are not met and the overall result has to lead to the negative validation (NOK). The value of the virtual sensor is set to zero ($VS1 = 0$), as shown by Figure 14 and the virtual sensor cannot provide the validated result.

$$VS1 = y_{2003} = F_{2003}(x_1, x_2, x_3) = 0. \quad (23)$$

The experimental validation of our model was also extended for digital sensors with the logic signal. Figure 15 shows the simulation of the digital sensors with logic values:

- $PS_1: x_1 = 1$;

- PS2: $x_2 = 0$;
- PS3: $x_3 = 1$.

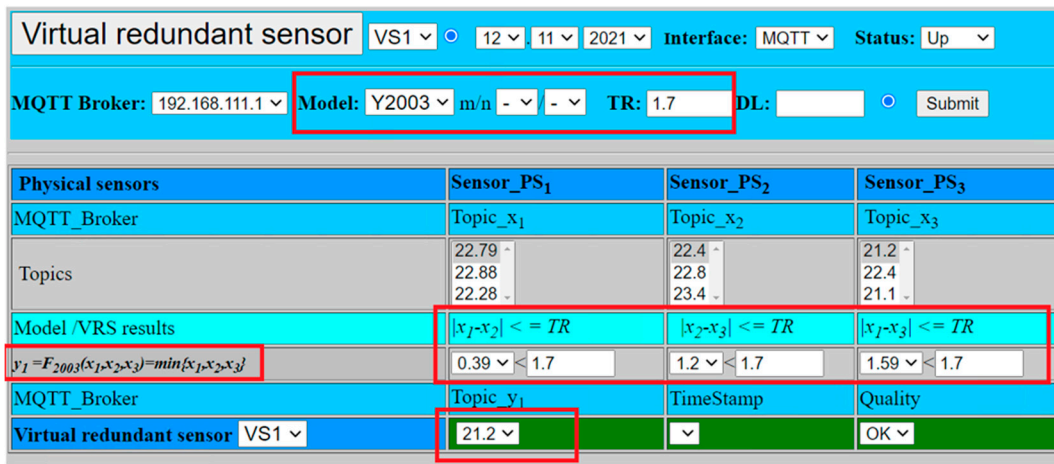


Figure 14. Virtual redundant sensor with fusion function F_{2003} and positive result.

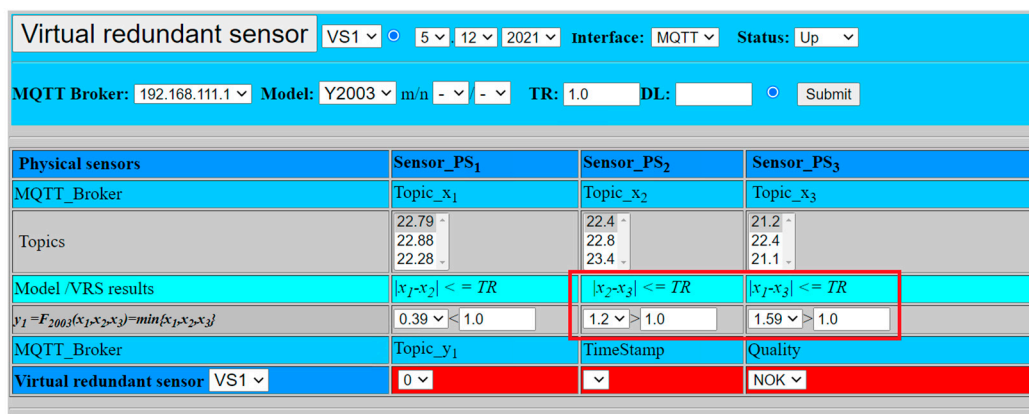


Figure 15. Virtual sensor with fusion function F_{2003} and negative result.

The result is calculated according to Formula (14):

$$y_{2003} = F_{2003}(x_1, x_2, x_3) = 1. \tag{24}$$

In addition, the fusion function F_{2003} with voting gives the same result VS1 = 1, because two from three sensors provide the same signal log. 1, as shown in Figure 16.

The extended model of virtual redundant sensor was also experimentally verified regarding its redundancy and high-availability features. The second virtual sensor VS12, with open MQTT connections to all physical sensors and application, was activated in addition to VS11. The virtual sensor model with duplexing was selected for an experimental verification, due to necessity to deal with safety-related applications requiring very low T_{ST} (Figure 7). The experiment was limited just to fusion function Y2002 with the aim to explore redundancy with duplexing of two virtual sensors VS11, VS12, using the same values from physical sensors as in previous experiments.

VS11 virtual sensor validated result from two physical sensors. The selected fusion function provided as a result the minimum from physical sensors values (22.4), as shown in Figure 17. The result was sent via AMQP to VS12 virtual sensor for result alignment (25), which was checked by VS12 and confirmed (26). After confirmation, both results were published for application by two MQTT brokers.

$$y_1 = \{R_{id}, T_{VS11}, p\} = \{22.4, 12.1.2022\ 11 : 53 : 13\ PM, 01082022, 0\}, \tag{25}$$

$$y_2 = \{R_{id}, T_{VS12}, p\} = \{22.4, 12.1.2022\ 11 : 53 : 14\ PM, 01082022, 0\}. \quad (26)$$

Virtual redundant sensor VS1 5 12 2021 Interface: MQTT Status: Up

MQTT Broker: 192.168.111.1 Model: Y2003 m/n TR: 0 DL: Submit

Physical sensors	Sensor_PS ₁	Sensor_PS ₂	Sensor_PS ₃
MQTT_Broker	Topic_x ₁	Topic_x ₂	Topic_x ₃
Topics	1 1	0 1	1 1
Model /VRS results	x ₁ x ₂	x ₂ x ₃	x ₁ x ₃
$y_1 = F_{2003}(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3$	0	0	1
MQTT_Broker	Topic_y ₁	TimeStamp	Quality
Virtual redundant sensor VS1	1		OK

Figure 16. Virtual sensor with digital sensors and fusion function F_{2003} .

Virtual redundant sensor VS11 18 1 2022 Interface: MQTT Status: Up

MQTT Broker: 192.168.111.1 Model: Y2002 m/n TR: 0.4 DL: Submit

Physical sensors	Sensor_PS ₁	Sensor_PS ₂	Sensor_PS ₃
MQTT_Broker	Topic_x ₁	Topic_x ₂	Topic_x ₃
Topics	22.79 22.88 22.28	22.4 22.8 23.4	0 0 0
Model /VRS results	$x_1x_2 \leq TR$		
$y_1 = F_{2002}(x_1, x_2) = \min\{x_1, x_2\}$	0.39 < 0.4		
MQTT_Broker	Topic_y ₁	TimeStamp	Quality
VS11	22.4		NOK
Virtual redundant sensor with duplexing	Topic_y / x ₁ , x ₂	TimeStamp	Result alignment: R _{id} /p
VS11 (pair) AMQP	22.4 22.79 22.4	T _{VS11} 1/18/2022 11:24:57 PM	01082022 0
VS12 (pair) AMQP	0 - -	T _{VS12} -	- 0

Virtual redundant sensor VS11 18 1 2022 Interface: MQTT Status: Up

MQTT Broker: 192.168.111.1 Model: Y2002 m/n TR: 0.4 DL: Submit

Physical sensors	Sensor_PS ₁	Sensor_PS ₂	Sensor_PS ₃
MQTT_Broker	Topic_x ₁	Topic_x ₂	Topic_x ₃
Topics	22.79 22.88 22.28	22.4 22.8 23.4	0 0 0
Model /VRS results	$x_1x_2 \leq TR$		
$y_1 = F_{2002}(x_1, x_2) = \min\{x_1, x_2\}$	0.39 < 0.4		
MQTT_Broker	Topic_y ₁	TimeStamp	Quality
VS11	22.4		NOK
Virtual redundant sensor with duplexing	Topic_y / x ₁ , x ₂	TimeStamp	Result alignment: R _{id} /p
VS11 (pair) AMQP	22.4 22.79 22.4	T _{VS11} 1/18/2022 11:24:57 PM	01082022 0
VS12 (pair) AMQP	22.4 22.79 22.4	T _{VS12} 1/18/2022 11:24:58 PM	01082022 0

Figure 17. Virtual redundant sensor VS11 with duplexing with example of result confirmation.

To check proper handling, the failure of VS12 virtual sensor has been simulated as well. In that case, VS11 did not get confirmation from VS12. VS11 consequently set parameter $p = 1$ and published the result value to MQTT broker (27). The setting of $p = 1$ notified application to use exclusively just VS11 values and ignore VS12 values, till recovery with parameter $p = 0$ is available again, as shown Figure 18.

$$y_1 = \{R_{id}, T_{VS11}, p\} = \{22.4, 12.1.2022\ 11 : 53 : 13\ PM, 01082022, 1\}. \quad (27)$$

Virtual redundant sensor			
VS11	18	1	2022
Interface: MQTT	Status: Up		
MQTT Broker: 192.168.111.1	Model: Y2002	m/n	TR: 0.4 DL: Submit
Physical sensors	Sensor_PS ₁	Sensor_PS ₂	Sensor_PS ₃
MQTT_Broker	Topic_x ₁	Topic_x ₂	Topic_x ₃
Topics	22.79 22.88 22.28	22.4 22.8 23.4	0 0 0
Model \VRS results	$ x_1 - x_2 \leq TR$		
$y_1 = F_{2002}(x_1, x_2) = \min\{x_1, x_2\}$	0.39		
MQTT_Broker	Topic_y ₁	TimeStamp	Quality
VS11	22.4		NOK
Virtual redundant sensor with duplexing	Topic_y / x1, x2	TimeStamp	Result alignment: R _{id} /p
VS11 (pair) AMQP	22.4 22.79 22.4	T _{VS11} 1/18/2022 11:27:50 PM	01082022 1
VS12 (pair) AMQP	0 - -	T _{VS12} -	- 0

Figure 18. Virtual redundant sensor VS11 with duplexing and example of VS12 failure.

6. Conclusions

In this article, we share our research results concerning the virtual redundant sensors, which are intended for SRCSSs. The main idea was to extend and enrich the capabilities of physical sensors by additional information with focus on so-called RAMS parameters. Virtual redundant sensors with different safety and reliability properties can be created from a set of physical sensors. Depending on the specification of the requirements for RAMS parameters of specific application, the appropriate combination of physical sensors and the mathematical model can be selected. Our focus was paid on the creation and validation of the appropriate numerical models which would define fusion functions for those parameters. Having combined physical sensors values, we are able to apply various models to influence overall results and contribute to RAMS parameters. In addition, our model was extended in order to ensure the required redundancy of virtual sensors and provide high-availability solution to safety-related applications. Two approaches based on mirroring and duplexing of virtual sensors were elaborated.

In order to validate our proposed approach, we have chosen the simpler models (*moon*) with fusion functions F_{2002} and F_{2003} . The virtual redundant sensors provided enriched value, which is calculated as the minimum (based on the requested safety level) or the mean value (based on the requested availability) from physical sensors' values. This approach helped us to simulate the needed capabilities of virtual redundant sensors even with changing values in the defined time frame. Experimental evaluation was based on emulation of physical sensors connected to MQTT broker (Rabbit MQ), which was connected with virtual sensor instance in both roles of MQTT subscriber and publisher. The physical and virtual sensors had WEB-based graphical user interface (ASP.NET) to select the needed numerical model and its parameters (TR) to validate results within the defined tolerance range or decision level. Our implemented experimental solution supports both implementation scenarios (Cloud based, Edge based). The experimental solution utilized AMQP protocol for the extended model with duplexing of virtual sensors, result alignment

(R_{id}) and verification (p). The used products (MQTT broker) and platforms can be chosen according to specific use-cases and required parameters.

The results of testing have shown that the proposed virtual redundant sensor concept can be applied to improve RAMS parameters of physical sensor. Our fusion functions calculated results according to selected model in all cases. The proposed model can handle also failure of one or more physical sensors. Not standard cases were marked with quality status “Not OK” and result value was set according to model to 0, as documented by attached GUI screens of the virtual redundant sensor. The extended model with duplexing was fully verified and embedded redundancy enabled virtual sensor operation without interruption of delivery validated values when one of the virtual sensors failed, so overall solution confirmed requested virtual redundant sensor capabilities.

All in all, the proposed concept of the virtual redundant sensor can provide additional value to safety-relevant systems, simplifying control application by validating sensors results and providing higher level of safety level and availability for safety related processes. However, this conceptual work just justifies the proposal itself. For real applications, we would have to consider a more complex definition of the fusion functions, because our used minimum/average values can be used for specific processes, but do not represent generally all processes (different logic required by process). Therefore, our future work could be oriented to incorporate flexible creation of fusion functions in form of library, or via “plugin” concept. As well as there is a potential to extend the model for dynamic changes of sensor values, that have not been simulated in our experiment. With respect to extended model, the duplexing approach requires physical sensors and application to have dual MQTT connection to both virtual sensors. This precondition might not be possible in all use-cases. Therefore, mirroring concept and its experimental verification shall be elaborated as well in future work, mainly if T_{ST} parameter can be accepted by safety-related application.

Author Contributions: The work presented in this article was carried out in collaboration with all authors. P.P. defined the research topic, guided the research goals, performed the experiments. K.R. proposed a mathematical model and reviewed the article. E.B. and A.K. wrote the theoretical part, edited, and reviewed the article. All authors have read and agreed to the published version of the manuscript.

Funding: The publication was funded by the Cultural and Educational Grant Agency MŠVVaŠ SR, grant number 008ŽU-4/2021: Integrated Teaching for Artificial Intelligence Methods at the University of Žilina.

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

References

1. Peniak, P.; Bubeníková, E.; Kanáliková, A. The virtual redundant sensors via Edge Computing. In Proceedings of the 2021 International Conference on Applied Electronics (AE), Pilsen, Czech Republic, 7–8 September 2021; pp. 1–5. [CrossRef]
2. Madakam, S.; Ramaswamy, R.; Tripathi, S. Internet of Things (IoT): A Literature Review. *J. Comput. Commun.* **2015**, *3*, 154–196. [CrossRef]
3. Xing, L.; Zhao, G.; Wang, Y.; Mandava, L. Competing failure analysis in IoT systems with cascading functional dependence. In Proceedings of the 2018 Annual Reliability and Maintainability Symposium (RAMS), Reno, NV, USA, 22–25 January 2018; pp. 1–6.
4. Huangshui, H.; Guihe, Q. Fault management frameworks in wireless sensor networks. In Proceedings of the 2011 Fourth International Conference on Intelligent Computation Technology and Automation, Shenzhen, China, 28–29 March 2011; Volume 2, pp. 1093–1096.
5. Qikun, S.; Bin, J.; Peng, S. *Fault Diagnosis and Fault-Tolerant Control Based on Adaptive Control Approach*; Springer: New York, NY, USA, 2017; Available online: <https://link.springer.com/content/pdf/10.1007/978-3-319-52530-3.pdf> (accessed on 11 November 2021).
6. Zhang, L.; Yang, G.-H. Observer-Based Adaptive Decentralized Fault-Tolerant Control of Nonlinear Large-Scale Systems with Sensor and Actuator Faults. *IEEE Trans. Ind. Electron.* **2019**, *66*, 8019–8029. [CrossRef]
7. Vachálek, J.; Šišmišová, D.; Vašek, P.; Fiřka, I.; Slovák, J.; Šimovec, M. Design and Implementation of Universal Cyber-Physical Model for Testing Logistic Control Algorithms of Production Line’s Digital Twin by Using Color Sensor. *Sensors* **2021**, *21*, 1842. [CrossRef]

8. Janota, A.; Nemeč, D.; Hrubaš, M.; Pirník, R. Knowledge-based approach to selection of weight-in-motion equipment. In *Challenge of Transport Telematics, Proceedings of the 16th International Conference on Transport Systems Telematics, TST 2016, Katowice-Ustroń, Poland, 16–19 March 2016*; Mikulski, J., Ed.; Springer International Publishing AG: Cham, Switzerland, 2016; Volume 640, pp. 1–12.
9. Nemeč, D.; Hrubaš, M.; Šimák, V. Intelligent real-time MEMS sensor fusion and calibration. *IEEE Sens. J.* **2016**, *16*, 7150–7160. [[CrossRef](#)]
10. Madria, S.; Kumar, V.; Dalvi, R. Sensor Cloud: A Cloud of Virtual Sensors. *IEEE Softw. J.* **2014**, *31*, 70–77. Available online: <https://ieeexplore.ieee.org/abstract/document/6662351> (accessed on 11 November 2021). [[CrossRef](#)]
11. Jiang, Y.; Yin, S.; Dong, J.; Kaynak, O. A Review on Soft Sensors for Monitoring, Control, and Optimization of Industrial Processes. *IEEE Sens. J.* **2021**, *21*, 12868–12881. [[CrossRef](#)]
12. Šabanovič, E.; Kojis, P.; Šukevičius, Š.; Shyrokau, B.; Ivanov, V.; Dhaens, M.; Skrickij, V. Feasibility of a Neural Network-Based Virtual Sensor for Vehicle Unsprung Mass Relative Velocity Estimation. *Sensors* **2021**, *21*, 7139. [[CrossRef](#)] [[PubMed](#)]
13. Paepae, T.; Bokoro, P.N.; Kyamakya, K. From Fully Physical to Virtual Sensing for Water Quality Assessment: A Comprehensive Review of the Relevant State-of-the-Art. *Sensors* **2021**, *21*, 6971. [[CrossRef](#)] [[PubMed](#)]
14. Pedrollo, G.; Konzen, A.A.; de Moraes, W.O.; Pignaton de Freitas, E. Using Smart Virtual-Sensor Nodes to Improve the Robustness of Indoor Localization Systems. *Sensors* **2021**, *21*, 3912. [[CrossRef](#)] [[PubMed](#)]
15. Brunello, A.; Urgolo, A.; Pittino, F.; Montvay, A.; Montanari, A. Virtual Sensing and Sensors Selection for Efficient Temperature Monitoring in Indoor Environments. *Sensors* **2021**, *21*, 2728. [[CrossRef](#)] [[PubMed](#)]
16. Li, Z.; Ye, J.; Wu, H. A Virtual Sensor for Collision Detection and Distinction with Conventional Industrial Robots. *Sensors* **2019**, *19*, 2368. [[CrossRef](#)]
17. Guzmán, C.H.; Carrera, J.L.; Durán, H.A.; Berumen, J.; Ortiz, A.A.; Guirette, O.A.; Arroyo, A.; Brizuela, J.A.; Gómez, F.; Blanco, A.; et al. Implementation of Virtual Sensors for Monitoring Temperature in Greenhouses Using CFD and Control. *Sensors* **2019**, *19*, 60. [[CrossRef](#)] [[PubMed](#)]
18. Schmidt, S.; Oberrath, J.; Mercorelli, P. A Sensor Fault Detection Scheme as a Functional Safety Feature for DC-DC Converters. *Sensors* **2021**, *21*, 6516. [[CrossRef](#)] [[PubMed](#)]
19. Gao, L.; Li, D.; Li, D.; Yao, L.; Liang, L.; Gao, Y. A Novel Chiller Sensors Fault Diagnosis Method Based on Virtual Sensors. *Sensors* **2019**, *19*, 3013. [[CrossRef](#)] [[PubMed](#)]
20. Mattera, C.G.; Quevedo, J.; Escobet, T.; Shaker, H.R.; Jradi, M. A Method for Fault Detection and Diagnostics in Ventilation Units Using Virtual Sensors. *Sensors* **2018**, *18*, 3931. [[CrossRef](#)] [[PubMed](#)]
21. Rástočný, K.; Žďánský, J. Specificities of safety PLC based implementation of the safety function. In *Proceedings of the International Conference on Applied Electronics, AE, Pilsen, Czech Republic, 5–7 September 2012*; pp. 229–232, ISBN 978-80-261-0038-6. ISSN 1803-7232.
22. Dekking, F.M.; Kraaikamp, C.; Lopuhaä, H.P.; Meester, L.E. *A Modern Introduction to Probability and Statistics (Understanding Why and How)*; Springer: London, UK, 2006; ISBN 978-1-84628-168-6.
23. *EN61508*; Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems. IEC: Geneva, Switzerland, 2010.
24. OASIS. Specification MQTT Version 3.1.1. Available online: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html> (accessed on 11 November 2021).
25. Egli, P.R. MQ Telemetry Transport—An Introduction to MQTT, A Protocol for M2M and IoT Applications. Available online: http://www.indigoo.com/dox/wsmw/1_Middleware/MQTT.pdf (accessed on 11 November 2021).
26. ISO/IEC. ISO/IEC 20922:2016, Information Technology—Message Queuing Telemetry Transport (MQTT) v3.1.1. Available online: <https://www.iso.org/standard/69466.html> (accessed on 11 November 2021).
27. RabbitMQ. RabbitMQ Is the Most Widely Deployed Open Source Message Broker. Available online: <https://www.rabbitmq.com> (accessed on 11 November 2021).
28. RabbitMQ. RabbitMQ Tutorial. Available online: <https://www.tutlane.com/tutorial/rabbitmq> (accessed on 11 November 2021).