JOURNAL OF AIRCRAFT
Vol. 47, No. 3, May–June 2010

Fusion of Smart-Sensor Standards and Sensors with Self-Validating Abilities

Pavel Pačes,* Michal Reinštein,† and Karel Draxler‡
Czech Technical University in Prague, 166 27 Prague, Czech Republic

DOI: 10.2514/1.43735

Airborne applications require a high degree of reliability, which is typically ensured by development guides, testing, quality checking, and overall certification processes. Although these processes provide a high level of product safety and reliability, the electronic devices can fail for various reasons. One of the main present-day problems is incompatibility of the communication interfaces of the smart sensors. This paper proposes a way of using a standardized IEEE 1451 interface with the information necessary for sensor self-validation ability. The necessary data are saved within the memory of the extended Transducer Electronics Data Sheet standard. This paper presents usage of the extended Transducer Electronics Data Sheet data on a servomechanism actuator with a feedback loop (servomechanism), designed for an unmanned aircraft.

I. Introduction

THE safe operation of sensors and actuators is primarily ensured by their redundancy together with a voter device that marks validity of the output signal. Technical development has made highly integrated electronics inexpensive and available for various applications; this is also true for the area of sensing devices. A smart sensor is usually a device consisting of a sensor element, analog data processing, analog-to-digital (AD) conversion, digital processing, and a digital output interface. Today, smart sensors are in use all around us, ready for measuring temperatures and for interconnection with testing devices, monitoring systems, systems of intelligent buildings, etc. All of these systems suffer from various digital output interfaces that differ in physical layers, logical levels, and communication algorithms. The group of IEEE 1451 standards proposes a way to standardize interfaces of smart sensors. First versions of the IEEE 1451 standard defined a new physical interface with its own logical levels and communication protocol. Because of the number of new emerging standards, the IEEE 1451 proposed interface was not widely spread. Despite of all changes, IEEE 1451 still divides the smart sensor at the transducer interface module (TIM) and network-capable application processor (NCAP), where TIM represents a sensing element and NCAP represents a gateway between a group of TIMs into a higher system. The important thing introduced by IEEE 1451 is electronic information about all details related to the sensor that is available in the nonvolatile memory. These data are collected in the Transducer Electronics Data Sheet (TEDS), which contains information about manufacturer, measured value, units, date of calibration, and calibration curve that can be saved in a number of ways (function and lookup table). A data format is predefined for sensors and referenced like a channel with assigned Channel TEDS. Calibration data related to a channel are saved in Calibration TEDS. The standard IEEE 1451.0 , the last issue from 2007, introduces a set of services for the TEDS and information manipulation. The physical layers are not being newly designed at all, but existing networks are used and included into the group of IEEE 1451 standards (e.g., wireless LAN and radio-frequency identification). More description about IEEE 1451 is presented in [2,3]. Controller–Area Network (CAN) is widely used in the automotive industry; its physical layer has no standard number yet in the group of IEEE 1451 standards.

Smart systems are used in aeronautics board instrumentation, but there is no IEEE 1451 standard. The standard is primarily suitable for measurements related to proving aircraft airworthiness [4], which includes a lot of sensors that need to be interconnected quickly.

The method proposed in this paper takes the existing idea of TEDS information and extends it in order to provide capability to self-validate measured values and also to provide information recognized from a measured signal. Common attributes were identified, and algorithms that can be used as an add-on safety feature for future sensing and actuating devices were developed. These common attributes are later used as building blocks for reusable software objects, in conjunction with the standards for smart-sensor interconnection. This paper proposes the use of extended smart devices on an unmanned aircraft (UA) (Fig. 1) as a standardized approach to simplify and improve their usage and maintenance through better data availability.

II. Device Under Development

The proposed methods were tested on a group of servomechanisms (SMs) (Fig. 2) that controls the Mamok UA that is being developed in the Czech Republic as a modernized replacement for the Sojka airplane. Each actuator converts an electric signal to a mechanical movement that controls such components as the engine power lever, the rudder, etc. The electric input signal is represented as a CAN-bus datagram, processed by the central module unit and converted into an angular movement. The distributed digital control simplifies the control, lowers the price, brings down the total weight, and improves reliability of the transferred commands. There are nine of these servomechanism modules on every UA (Fig. 3). SMs provide control of altitude (Fig. 3c), direction of flight (Figs. 3b and 3d), aircraft tilt (Figs. 3a and 3e), wheel braking (Fig. 3g), engine power lever (Fig. 3f), and ground turning (Fig. 3h).

The servomechanism can be divided into two parts: the electronic control system and the mechanical part containing the engine and gearbox (Fig. 4). The figure shows a simple design containing power conditioning, the main microprocessor (Philips LPC2129) with connected peripherals as memory for TEDS data, CAN driver, and engine drivers. The final assembly is shown in Fig. 5. The mechanical part of the servomechanism is based on the Hitec HS-5955TG

Presented at the 27th IEEE/AIAA Digital Avionics System Conference, St. Paul, MN, 26–30 October 2008; received 11 February 2009; revision received 12 September 2009; accepted for publication 20 January 2010.

*Ph.D. Student, Faculty of Electrical Engineering, Department of Measurement, Technicka 2, Praha 6; pacesp@feld.cvut.cz. Student Member AIAA.

†Ph.D. Student, Faculty of Electrical Engineering, Department of Measurement, Technicka 2, Praha 6; reinsm1@feld.cvut.cz. Student Member AIAA.

‡Associate Professor, Faculty of Electrical Engineering, Department of Measurement, Technicka 2, Praha 6; draxler@feld.cvut.cz.

Fig. 1 Mamok UAV.

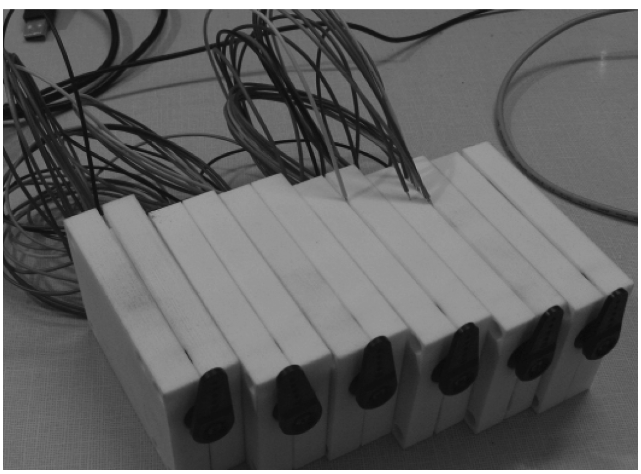

Fig. 2 Group of servomechanisms.

robotic servo, in which the original control system was removed and replaced by a new design that enables further software enhancement that could not be achieved with the original electronic interface. Software performs SM control algorithm and input–output control of the engine, read–write operation from external memory, communication sequences, and watchdog services.

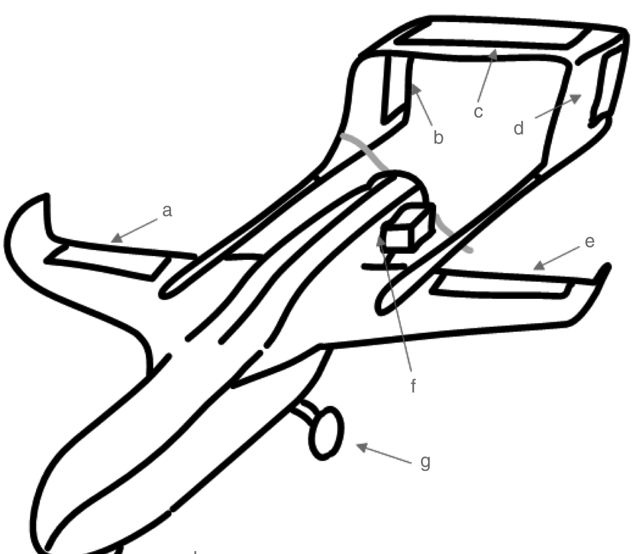

Fig. 3 Placement of servomechanisms.

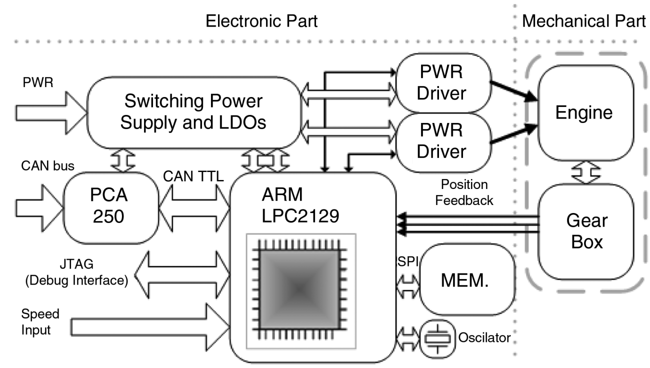

Fig. 4 Servomechanism block diagram.

Fig. 5 Final actuator assembly.

III. TEDS Extension

The main aim is to develop a feasible and easily maintained device in which different presumptions are placed together. These presumptions are coded as TEDS data included in the all devices within a network (Fig. 6). TEDS data ether need to be extended or they are composed from existing IEEE 1451 blocks or other standards are employed. The presented proposal takes advantage of the following items:

1) Existing description of SM feedback data representing the servomechanism position: channel TEDS.

2) The mechanism for calibrating the output value has already been defined: calibration TEDS.

3) A standardized command interface for the actuator exists: CANaerospace.

4) An application programming interface is provided for information access: IEEE 1451.0 [1].

5) An event generation mechanism is provided: IEEE 1451.0 [1].

Today's microprocessors have enough computing power to perform tasks such as sensor sensing, digital processing, and time measurement; they also can add calculations that allow catching important parts of the measured signal. Next, we will discuss two simple methods for direct signal output validation without external support. The following methods can be designated as IEEE 1451 user-defined TEDS structures: 1) range/limit check (Fig. 7a), 2) magnitude jump and rate of change (Fig. 7b), 3) magnitude model check (Fig. 7e), and 4) magnitude prediction (Fig. 7c).

The range signal validation is shown in Fig. 7a, which also shows signal-validation-block validity output. To describe the signal inside its defined range is complicated because of the unknown reference signal and its behavior. The most difficult method is magnitude prediction, which compares the actual measurement with the known point of the value on its transfer characteristic. Not all applications

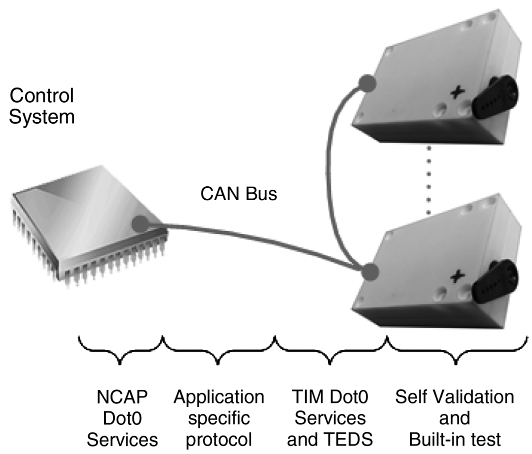

Fig. 6 Servo connection and services.

need to use all of these techniques; hence, unused TEDS proposals can be omitted.

IEEE 1451.0 [1] defines three devices: sensors, actuators, and event sensors. An event sensor is a simple block that generates an event when the analog input crosses a defined threshold. The IEEE 1451 standard consists of a short description of the sequence that leads to generation of an event. This paper describes simple input-signal-validation techniques that are applicable to all signals. We assume that the sensor knows exactly what it measures. It knows that the measured signal can reach only a defined range of values, and the rate of change of the signal is only within a defined range. All of these values were incorporated into the TEDS structures. The sensor also knows its actual position on the sensor's transfer characteristics and can compare its value with other sensors or with the model of an input signal.

The signal path from the sensing element, through signal conditioning and A/D conversion to the microprocessor, gives numerical values that are later processed by software, and into that we add another software block that contains a signal-validation gate, a fault-accommodation block, and fault-event generation. The data flow can be described as an input value that passes into the signal-validation (SV) block (see Fig. 8), which calculates requested data from the signal according to IEEE 1451.0 [1] TEDS (Channel, User-Defined TEDS, etc.) with the help of SV TEDS and system time services. The data processing results describe the validity or invalidity of the input signal.

Figure 9 shows the data flow of the software controller in the servomechanism. An analog value is read and then validated by the signal-validation block. Subsequent system behavior depends on the value that is returned by the SV block. This output value depends on whether or not an anomaly was detected and where we assume a logical output signal. In the case of an invalid return value, some other means of anomaly accommodation has to be performed by the system. For example, fault accommodation of a blocked gearbox results in an immediate stop of the engine driving signal. A blocked

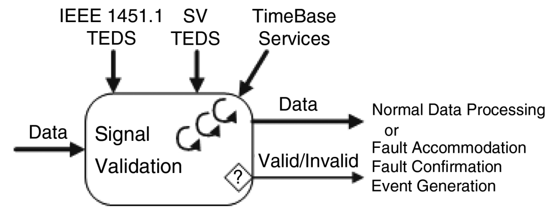

Fig. 8 SV block.

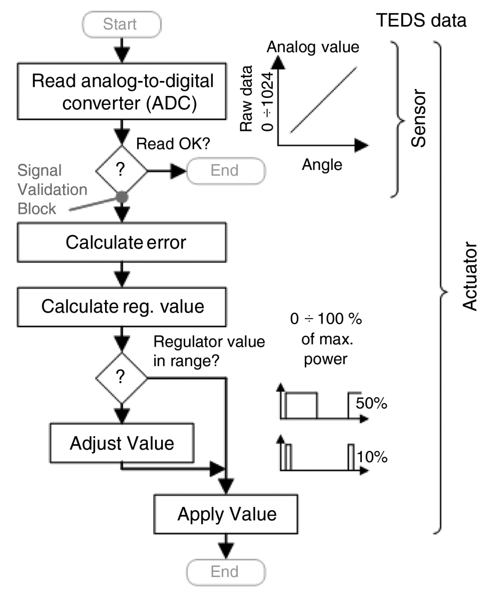

Fig. 9 SV-block placement (ADC is the analog-to-digital converter).

gearbox is detected when the engine is under full power, but the feedback value does not change for the specific amount of time. Generally, fault accommodation includes data measured by a group of the sensors measuring the same quantity, statistically calculated data, or model-based comparison. In the described application we use known attributes of the measured value and a model of the device that was generated from step responses. The model is used in a Kalman filter algorithm for prediction of the next measured value.

A. Range Check

The first and simplest signal-validation technique is a range test that determines whether or not the measured value is in a defined range of values. The range limitation can be caused by a physical quantity characteristic or by processing-path characteristics. Actually, the lower and upper signal limits are a part of the transducer channel TEDS [1]. Nevertheless, for the range-check purposes, the user-defined TEDS structure is depicted in Table 1, in which the group item indicates the TEDS borders. The second row

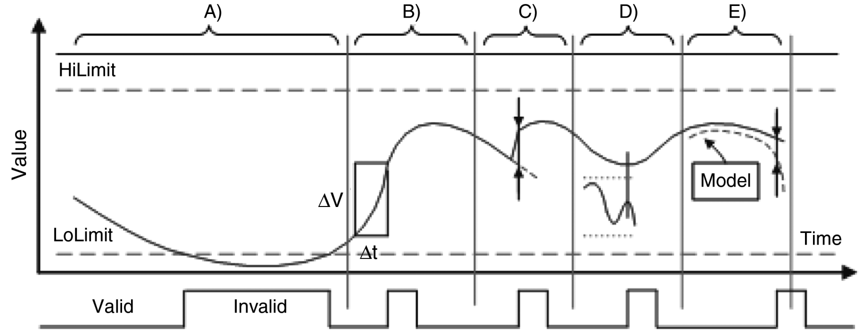

Fig. 7 Validation techniques.

Table 1 Limits TEDS structure

Field name	Description	Data type	No. octets
	Length	UInt32	4
TEDSID	TEDS identification header	UInt8	4
MaxLim	Number of records	UInt8	1
LimGrp	Limit group	——	——
ChID	Channel ID	Uint16	2
HiLim	High limit	Float32	4
LoLim	Low limit	Float32	4
	Checksum	UInt16	2

Table 2 Rate-of-change TEDS structure

Field name	Description	Data type	No. octets
	Length	UInt32	4
TEDSID	TEDS identification header	UInt8	4
MaxRate	Number of records	UInt8	1
RateGrp	Rate group	——	——
ChID	Channel ID	Uint16	2
MaxRate	Max signal rate change	Float32	4
	Checksum	UInt16	2

(TEDSID) defines the purpose of this TEDS, and the following line contains the number of range-check records. The limit group is a combination of limits and a channel ID item that connect the limit group with the transducer channel TEDS.

B. Rate of Change and Detection of Magnitude Jump

The input signal rate of change is described by extended TEDS values necessary for detection of the specified signal change in time. The technique requires hardware and software support for time-measurement services. A simple definition of rate-of-change TEDS is shown in Table 2. This TEDS consists of channel assignment (ChID) and maximal allowed change (MaxRate). The MaxRate field units are assumed to be in input signal units per second, and signal units are part of the transducer channel TEDS. This technique is suitable for magnitudes with slow changes, such as engine temperature measurements or altitude measurements. Current implementation uses moving-average filtering that cuts off peaks in signal. For future usage, where a different method of signal filtering could be used, the rate-of-change TEDS will require further development. The magnitude jump (Fig. 7c) is detected in a similar way, in which differences between specified samples are detected with no filtering algorithm. The simplest TEDS providing data for this purpose is also shown in Table 2.

C. Magnitude Prediction

Several approaches can be chosen to solve the general estimation problem. Given the appropriate vector of observations Z of size $(m \times 1)$, the vector of parameters to be determined X of size $(n \times 1)$ and assuming the model in the form of $Z = HX$, with H of size $(m \times n)$, the system is overdetermined for $m > n$ with enough information (equations) to specify all elements of X was chosen, but further methods have to be applied to guarantee a perfect data fit. One of the most common approaches in this case, which corresponds to our SM case, is to fit the data into the least-squares sense, as described by the equation

$$\hat{\mathbf{X}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{Z}; |\mathbf{H}^T\mathbf{H}| \neq 0 \tag{1}$$

There are, however, several drawbacks, as described by Grewal and Andrews [5], when using a simple least-squares estimation:

1) Results are predicated upon an assumed model, and miss-modeling can cause a serious flaw.

2) All data residuals, i.e., differences between predicted and measured values, are weighted equally; hence, there is no way to consider anomalous data.

3) There is no way to incorporate information regarding a priori knowledge of used parameters.

4) Batch processing is implied; all data need to be collected at first.

5) The criterion of the least-squares is data-fitting, not minimizing the estimation error.

To deal with all of the above concerns, a Kalman filter (KF) is the logical step to take. It brings into consideration points 2 to 5, so only the modeling still remains a problem. When compared with classical least-squares, the Kalman filter yields approximately the same results if the initial uncertainty in X is large, the system is overdetermined or exactly determined, and all observations are of equal quality, and this almost never happens.

D. Kalman Filtering

According to Grewal and Andrews [5], Kalman filtering is primarily a procedure for combining noisy sensor outputs to estimate the state of a general system with uncertain dynamics: dynamics that need to be precisely modeled. The system state vector includes any variables of the system, as well as inner variables for modeling time-correlated noise sources and random sensor parameters. The actual model determines the complexity and computational load of the KF. To determine the final uncertainty of the estimated system states provided by the KF, a covariance analysis was performed. Covariance analysis is a part of the KF algorithm and can be performed even without real data, based only on the sensor noise parameters given by the manufacturer. In the end, it shows how much the estimated system states vary from the optimal values in the means of variance, assuming the Gaussian distribution.

E. Servomechanism Model Creation

To create a suboptimal mathematical model of the servomechanism that would be mathematically stable and create a manageable computational load, a proper approximation method was sought. As Nassar [6] described, there are several simple random processes that can be used to approximate noises entering the KF, such as random constant, random walk or exponentially correlated random process (the Gauss–Markov process of first order). These processes exhibit a

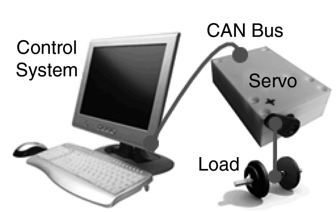

Fig. 10 Test setup.

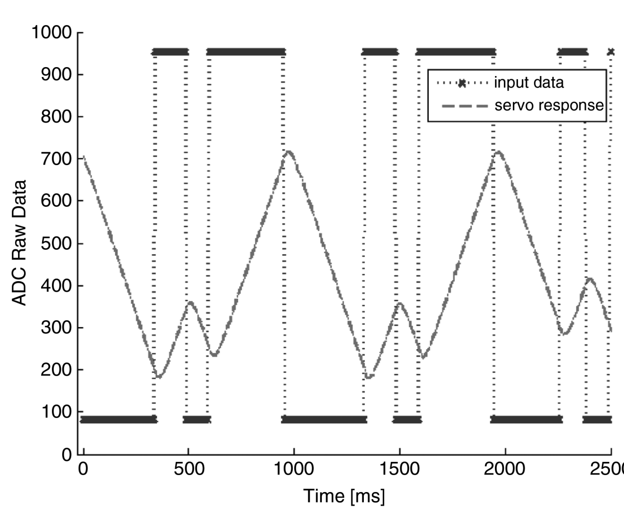

Fig. 11 Servo's pseudorandom input and its response.

Table 3 Model TEDS structure

Field name	Description	Data type	No. octets
	Length	UInt32	4
TEDSID	TEDS identification header	UInt8	4
ChID	Channel ID	Uint16	2
NumGrp	Numerator group	——	——
NumNu	Number of numerators	UInt8	1
NumItem	Numerator item no.	Float32	4
DenGrp	Denominator group	——	——
DenNu	Number of denominators	UInt8	1
DenItem	Denominator item no.	Float32	4
	Checksum	UInt16	2

simple power spectral density trend, which can be suitable for sensor bias or drift approximations; however, to approximate the servomechanism behavior, it is not suitable. One of the possible solutions can be found when using higher-order Gauss–Markov (GM) processes. Any GM process of any order can be represented using an autoregressive (AR) process of appropriate order [6]. The AR process of order p can be described using a pole-zero transfer function $H(z)$, where $X(z)$ is the z transform of the input $x(k)$, $Y(z)$ is the z transform of the output $y(k)$, and $\alpha_1, \alpha_2, \ldots, \alpha_p$ and β_0 are the AR process parameters in discrete time:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\beta_0}{1 + \sum_{n=1}^{p} \alpha_n z^{-n}} \tag{2}$$

$$y(k) = -\sum_{n=1}^{p} \alpha_n y(k-n) + \beta_0 x(k) \tag{3}$$

In the end, Burg's method was used for the creation of the servomechanism's AR model, which is described in [6]. Various data sets corresponding to servomechanism response to different loads were collected and used as input to the estimator implemented in MATLAB.

IV. Results

Data acquired during servomechanism development were measured using the measurement setup shown in Fig. 10. A personal computer with a Universal Serial Bus to CAN-bus converter was used for connection with the servomechanism, which is equipped with a CAN-bus interface. The SM is loaded with specific weight, which is moved by the servomechanism lever according to an input signal into regulator service. The servomechanism responses are sampled with a rate of about 170 Hz and transferred to the PC by the CAN bus with a speed of 1 MB (maximum for CAN bus). This method was chosen because the information collected in this way will be the only source available to the device in the final installation.

Figure 11 shows the pseudorandom signal for transfer characteristics determination (dotted line) with the servomechanism response (dashed line). Measured data were processed and modified in order to analyze the important details of the signal. The signal prediction is dependent on the system model, for which we use the data TEDS structure shown in Table 3. The model is assumed to be in polynomial form and calculated by SM's microcontroller. The numerator and denominator degrees are expressed by their count, followed by an array of coefficients. The KF algorithm was enhanced to detect the data outage. If data outage is detected, the calculation of residuals and state vector update step is omitted. Then a driving input to the system model is triggered using the precomputed Kalman gain in which KF works as a predictor. As the outage ends, the KF is switched back into filtering regime, calculating the residuals and updating the Kalman gain values. Variables necessary for algorithm calculation are not part of the tabled data.

The data acquired from the servomechanism are shown in Fig. 12. This figure shows response to the input signal that follows Fig. 11. To show important functions of the magnitude estimation, the input data were modified to simulate failure of the feedback input. In the case of the servomechanism, the failure of input signal leads to saturation of the analog-to-digital converter that will measure maximal input value. An example of the saturation is shown in Fig. 12b.

One of the important signal components is the beginning of the signal response, where the algorithm waits for its history (Fig. 12a). Figure 12c shows problem with a concave change of signal direction. It can be seen that the model-based estimation continues with the

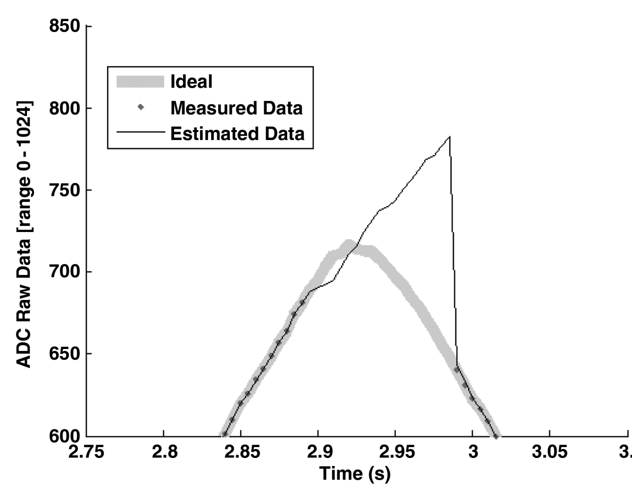

Fig. 13 Concave change: case C.

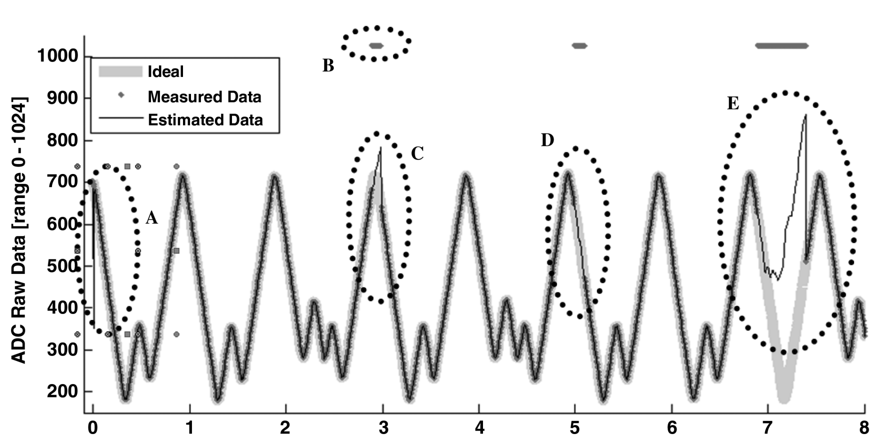

Fig. 12 Servomechanism response for random input sequence with data outage.

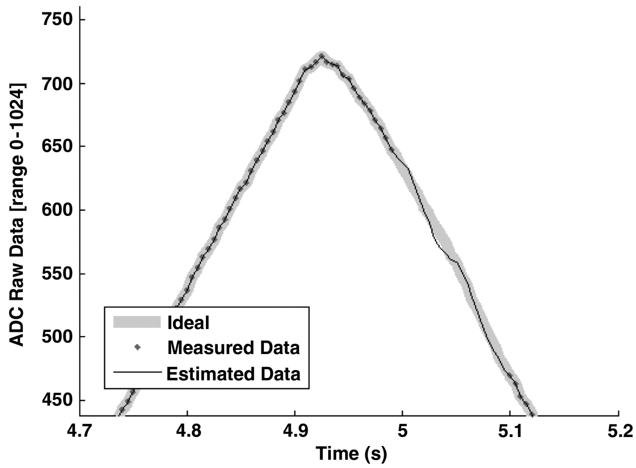

Fig. 14 Prediction: case D.

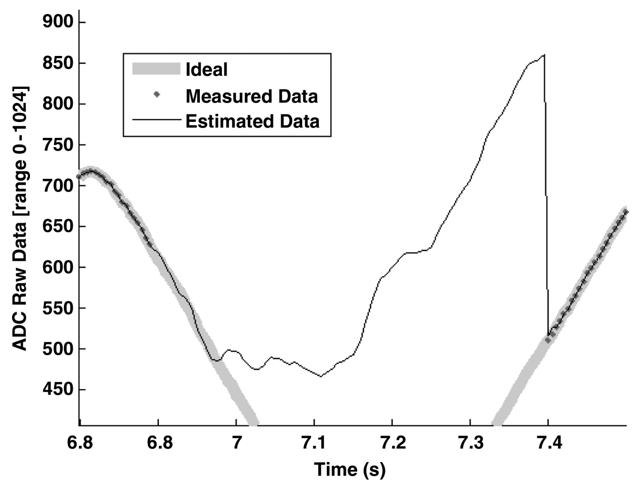

Fig. 15 Long-term nonstability: case E.

previous direction, which is incorrect, because the proper signal changed its direction. The prediction of the modeled signal can be improved by including the input command into the KF as an input data line (Fig. 11) and in detail (Fig. 12), but this is only applicable for feedback systems. The steep growth that returns estimated values back to the best line (Fig. 13) is caused by the return of proper feedback values.

The example in Fig. 12d (or details in Fig. 14) shows a situation in which the feedback signal is lost for a long time (0.12 s), which can be successfully covered by the data estimation. On the other hand, the longer data outage leads to unpredictable behavior, which is shown in Fig. 15.

V. Conclusions

A smart sensor, or a device called *intelligent* (see the definition of intelligence in [7]), should contain the following built-in features in a standardized form: 1) plug-and-play ability, 2) TEDS information availability in any form, 3) self-validation techniques, 4) fault-tolerance ability, and 5) sensor consciousness about measured value, as proposed in this paper.

The basic idea presented in this paper is to offer an existing solution (the TEDS data) and to extend it in order to be suitable for a new and reusable application. Already developed reusable algorithms are used in another application that allows suppressing bugs and picking up important parts of the measured signal. This paper proposes a new principle of using TEDS information [1] for signal validation that is used in a data processing algorithm of smart-sensor output. New TEDS structures are proposed as data storage for signal evaluation methods described in this paper. Described methods include signal limits (min, max) checking, signal rate-of-change calculation, and comparison of measured output with an estimated value. The third algorithm estimates a future output value with reference to previously measured data and compares this value with a value measured at the sensor input. In case of unacceptable differences, a time stamp is saved and a superior system is noted. A Kalman-filter-based algorithm is proposed and verified for servomechanism feedback output data filtering and estimation of a future value. The precision of prediction proves to be suitable for short-time-measurement outages. The algorithm's application is presented on an SM system for a new UAV project. The servomechanism combines a sensor and an actuator together with common and extended TEDS structures. Placement of the proposed algorithms is designed in the servomechanism control-loop feedback. The CAN-bus connection for data exchange with a master system is used in this paper.

Acknowledgments

This project was supported by the research program no. MSM6840770015, "Research of Methods and Systems for Measurement of Physical Quantities and Measured Data Processing," of the Czech Technical University in Prague, sponsored by the Ministry of Education, Youth and Sports of the Czech Republic. Pavel Pačes offers special thanks to Karel Draxler, who is the supervisor for this study, and to Michal Reinštein, the coauthor of this paper.

References

[1] "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats," Inst. of Electrical and Electronics Engineers, Piscataway, NJ, Sept. 2007.

[2] *How Can IEEE 1451 Be Applied?*, National Inst. of Standards and Technology, Gaithersburg, MD, http://ieee1451.nist.gov/, [retrieved 21 July 2008].

[3] Ramos, H. M., Ramos, P., and Pačes, P., "Development of a IEEE 1451 Standard Compliant Smart Transducer Network with Time Synchronization Protocol," *IMTC/2007 IEEE Instrumentation and Measurement Technology Conference Proceedings* [CD-ROM], Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 2007.

[4] Eccles, Lee H., "The Need for Smart Transducers: An Aerospace Test and Evaluation Perspective," *IEEE Instrumentation & Measurement Magazine*, Vol. 11, No. 2, April 2008, pp. 23–28.

[5] Grewal, M. S., and Andrews, A. P., *Kalman Filtering—Theory and Practice Using MATLAB*, Wiley-Interscience, New York, 2001.

[6] Nassar, S., *Improving the Inertial Navigation System (INS) Error Model for INS and INS/DGPS Applications*, Univ. of Calgary, Dept. of Geomatics Engineering, Rept. 20183, Calgary, Canada, 2003.

[7] Gottfredson, L. S., "Foreword to 'Intelligence and Social Policy'," *Intelligence*, Vol. 24, No. 1, 1997, pp. 1–12. doi:10.1016/S0160-2896(97)90010-6