Sensors and Actuators A 147 (2008) 623–632

Contents lists available at ScienceDirect

Sensors and Actuators A: Physical

journal homepage: www.elsevier.com/locate/sna

A novel fault-tolerant sensor system for sensor drift compensation

T.L. Chen*, R.Z. You

Department of Mechanical Engineering, National Chiao Tung University, Hsinchu 30056, Taiwan, ROC

ARTICLE INFO

Article history:
Received 31 August 2007
Received in revised form 14 April 2008
Accepted 25 May 2008
Available online 3 June 2008

Keywords:
Sensor drift
Real-time fault identification
Real-time fault compensation
Parity equations
Components redundancy

ABSTRACT

The conventional fault-tolerant sensor systems would fail when outputs from incorporated sensors are either noisy or drifting. This paper presents a novel real-time fault compensation method, which uses state estimation and compensation techniques, that the sensor system can perform robust measurements even when outputs from every incorporated sensor are noisy and drifting. In a simulation example, the proposed design can detect and correct the sensor errors (dc bias and drift) in real time. For the dc bias, the minimum detectable offset value is 0.1, which is the same as the standard deviation of the sensor noise. The compensated sensor output is biased at values smaller than 0.02. For the sensor drifts, the proposed method can compensate drifts for the change rate of drifts up to four times faster than that of the signal to be measured. The highest change rate of drifts, that can be compensated by this method, is determined by the standard deviation of the sensor noise.

1. Introduction

Sensor drifts can be found in lots of sensor applications and they may constitute a significant source of sensor unreliability. Currently, two main approaches are used to suppress the sensor drift. One approach identifies the source of the drift and then compensate for it [1,2]. Because many different underlying mechanisms can be responsible for generating the drift, this approach often requires the detail knowledge of sensor physics. The other approach assumes that the behavior of the drift can be characterized by several patterns. These patterns may be unknown beforehand but can be obtained by off-line learning. When these drifting sensors work online, their measurements are differentiated between correct signals and drifts by comparing them to the previously defined patterns [3–5]. Since this approach highly replies on the statistics of the sensor behavior, there exists a chance of false alarms.

A fault-tolerant sensor system is to maintain system output accuracy when some of its incorporated sensing elements are erroneous. This is achieved by finding the erroneous ones in a sensor array ("fault identification") and using the correct ones for the output of the sensor system. Many fault-tolerant sensor systems are constructed based on the "geometric redundancy" and "parity space approach," [6–8] which involve having identical sensors deployed at various locations so as to establish algebraic equations for each sensor output. These equations are referred to as either "parity equations" or "voting equations." The erroneous sensor unit can be identified by these equations and excluded from the sensor array.

For the parity space approach, the fault identification is often through the use of voting equations [6,7]. If the outputs from the incorporated sensor elements are contaminated by noise, with this approach, one must set up threshold values and an observation period. When outputs from the voting equations exceed the threshold value at the end of an observation period, a faulty sensor recognition is declared. The need for an observation period indicates that this approach cannot be done in real time [3]. As a consequence, the real-time "fault correction" method, which can be utilized for the sensor "self-repairing," is not attainable.

The fault-tolerant sensor system, achieved by the sensor redundancy design, can not be used when its incorporated sensors experience sensor drifts. This is because, due to the nature of the drift, every sensor in an sensor array can be drifting from time to time. And, after the first few drifting sensors are identified and excluded from the sensor array, the remaining sensor system would lose its redundancy to identify other drifting sensors. Therefore, in the case of sensor drifts, the fault-tolerant sensor system need to have the capability of self-repairing to maintain its redundancy and thus output accuracy.

This paper proposes a novel real-time fault compensation method for fault-tolerant sensor systems. Due to its capability of the real-time fault correction, this method can be applied to compensate sensor drifts. Different from other solutions to the sensor

* Corresponding author.
E-mail addresses: tsunglin@mail.nctu.edu.tw (T.L. Chen), witnessyo930@msn.com (R.Z. You).

0924-4247/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.sna.2008.05.026

drift, this method does not require the detail knowledge of sensor physics nor rely on the statistics of sensor behaviors. Thus, it can be useful to lots of sensor applications. This paper is organized as follows: the basics of the fault-tolerant theory are introduced in Section 2. The proposed real-time fault compensation method is shown in Section 3. The development of this algorithm starts from the real-time compensation for the dc offsets existed in sensor outputs, and then to tackle on the sensor drifts. The stability analysis and compensation accuracy of the proposed method are shown in Section 4. A case study and several simulation results are shown in Section 5. Section 6 discusses several unique features of this method. Section 7 concludes this paper.

2. Fault-tolerant theory

"Geometric redundancy" design is one of the most important designs for fault-tolerant systems. It starts with the following equation:

$$m = Hx + e \tag{1}$$

where, m is a vector of sensor measurements from a sensor array, x is the state vector to be measured by sensors, and e is the sensor noise with zero mean. Let H^* be the transpose conjugate of H and V be the null space of H^*, then

$$V^*H = 0 \tag{2}$$

The "parity equations," which are widely applied to fault-tolerant systems, are defined as follows:

$$\text{Parity Equations} \triangleq V^*m \tag{3}$$

Therefore, if all the sensors function properly, the outputs of parity equations are expected to be zero mean. As shown in previous researches [6–8], a system with n states normally needs $n+1$ sensors to form one parity equation, $n+2$ sensors to form two parity equations, and so forth. Furthermore, by assuming that only one faulty sensor exists in a sensor array, the system needs at least two parity equations to locate it.

These two parity equations are often converted into $n+2$ equations, each involving $n+1$ sensor outputs. That is to say, one can form a $(n+2)\times(n+2)$ matrix with its diagonal terms equal to zeros but non-zero elsewhere. The newly formed equations are referred to as "voting equations" with an associated "voting matrix" to distinguish them from the "parity equations." Furthermore, since the voting matrix is derived from two parity equations, its rank is two.

$$\text{Voting Equations} \triangleq C_{\text{voting}}m = C_{\text{voting}}\begin{bmatrix} m_1 \\ \vdots \\ m_{n+2} \end{bmatrix}$$

$$C_{\text{voting}} = \begin{bmatrix} 0 & c_{12} & \cdots & c_{1,n+2} \\ c_{21} & 0 & \cdots & c_{2,n+2} \\ \vdots & & & \vdots \\ c_{n+2,1} & \cdots & c_{n+2,n+1} & 0 \end{bmatrix} = \begin{bmatrix} C_{\text{vot}\,1} \\ C_{\text{vot}\,2} \\ \vdots \\ C_{\text{vot}\,n+2} \end{bmatrix} \tag{4}$$

where, $m_1,\ldots,m_{n+2}$ represent measurements from the $n+2$ sensor, $C_{\text{vot}\,1},\ldots,C_{\text{vot}\,n+2}$ are the row vectors of C_{voting}. A faulty sensor can be more easily identified from voting equations than from parity equations. For example, if the output of the first voting equation is zero but the rest are non-zeros, one can determine that sensor m_1 is faulty.

When sensor measurements are contaminated by noise, the above fault-finding method can not be done in real time because, at any time instant, none of the voting equation outputs is zero. Besides, the above method can not be used in the system when incorporated sensors experience sensor drifts. It is because every sensor in a sensor array can be drifting and thus the assumption of only one faulty sensor in a sensor array can hardly be satisfied.

3. A novel real-time fault compensation method

Here, a novel real-time fault compensation method is proposed for the case when outputs of incorporated sensors are contaminated by noise and signal drifts. This is done by formulating the conventional fault-identification method into a real-time state estimation and state compensation problem. The developments of this method are proceeded by three steps: real-time dc offset identification, real-time dc offset compensation, and real-time sensor drifts compensation.

3.1. Real-time dc offset identification

To identify dc offsets in sensor outputs using state estimation techniques, the dc offset of sensor j, d_j in Eq. (5), is separated from sensor measurements and modelled as a system state.

$$m_{\text{dc}j} = m_j + d_j, \quad j = 1,\ldots,n+2 \tag{5}$$

where, $m_{\text{dc}j}$ is the sensor output with dc offset and white noise while m_j is the sensor output with white noise. When these offsets are treated as system states, their "governing equations" can be written as:

$$\dot{d}_j = 0, \quad j = 1,\ldots,n+2 \tag{6}$$

Since the above system is static, from a system observability viewpoint [9], it needs $n+2$"output equations" to observe $n+2$states. The two parity equations, which were used to describe the relations between outputs of $n+2$ sensors, can be processed to obtain two output equations for the estimation. These two equations can be arbitrary chosen from two row vectors in the voting matrix, such as Z_p shown in Eq. (7).

$$Z_p \triangleq \begin{bmatrix} C_{\text{vot}\,1} \\ C_{\text{vot}\,2} \end{bmatrix}\begin{bmatrix} d_1 \\ \vdots \\ d_{n+2} \end{bmatrix} = \begin{bmatrix} C_{\text{vot}\,1} \\ C_{\text{vot}\,2} \end{bmatrix}\begin{bmatrix} m_{\text{dc}\,1} - m_1 \\ \vdots \\ m_{\text{dc}\,n+2} - m_{n+2} \end{bmatrix}$$

$$= \begin{bmatrix} C_{\text{vot}\,1} \\ C_{\text{vot}\,2} \end{bmatrix}\begin{bmatrix} m_{\text{dc}\,1} \\ \vdots \\ m_{\text{dc}\,n+2} \end{bmatrix}$$

$$Z_{\text{aux}} \triangleq C_{\text{voting}}de_id_i = \begin{bmatrix} C_{\text{vot}\,1}d & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & C_{\text{vot}\,n+2}d \end{bmatrix} d$$

$$= \begin{bmatrix} v_1 \\ \vdots \\ v_{n+2} \end{bmatrix} d = [\, d_1,\ldots,d_{n+2} \,]^{\text{T}}$$

The remaining output equations should come from the assumption that only one faulty sensor existed in the sensor array; this leads to the Z_{aux} equations shown in Eq. (7). In that equation, e_i is the unitary vector, and v_i is the fictitious noise required for the subsequent extended-Kalman-filter (EKF) signal processing.

With the system governing equations Eq. (6) and output equations Eq. (7), one can estimate these dc offsets in real time by constructing a state observer. Since the sensor noise is one of the major concerns, plus the associated output equations Z_{aux} are nonlinear, the EKF [10] is chosen to be the algorithm for the state observer.

The EKF algorithms were developed for discrete-time systems. When adopting EKF for this estimation system, the estimation algorithms are listed below:

$$\begin{aligned}
&P^{-}(k+1) = P(k)\\
&L(k+1) = P^{-}(k+1)H^{\mathrm{T}}(k+1)(H(k+1)P^{-}(k+1)H^{\mathrm{T}}(k+1) + R(k+1))^{-1}\\
&P(k+1) = (I - L(k+1)H(k+1))P^{-}(k+1)\\
&\hat{d}(k+1) = \hat{d}(k) + L(k+1)(z(k+1) - h(\hat{d}(k)))\\
&z(k) = [\, Z_p(k) \quad Z_{\mathrm{aux}}(k)\,]^{\mathrm{T}}, \qquad H(k+1) = \left.\frac{\partial}{\partial d}h(d)\right|_{d=\hat{d}(k)}
\end{aligned} \tag{8}$$

where, $\hat{d}$ is the estimated value of d; $L(k)$ is the observer gain at the k th sampling time; $P(k) \equiv E[(\hat{d}(k) - d(k))(\hat{d}(k) - d(k))^{\mathrm{T}}]$ is the state covariance matrix at the k th sampling time; R is the noise covariance matrix associated with output equations; $h(\cdot)$ is the output equations (Eq. (7)) for sensor measurements.

Note that there are other sets of output equations that can implement the constraint of only one faulty sensor in a sensor array, for example: the multiplication of any two offsets equals to zero ($d_i \times d_j = 0,\ i \neq j$). Different set of output equations leads to different convergence properties for state estimation. Our experiences indicate that, with other set of output equations, the resulting estimation system is likely to be "locally observable" [11]. Besides, those approaches require an excess amount of output equations and that increases numbers of local minimums in the state estimation. Thus, the estimated state values are likely to be wrong. The formulation of the Z_{aux} equations is the key of the proposed real-time fault compensation method.

3.2. Real-time dc offset compensation

Once the dc offset in each sensor output can be identified in real time, the offset compensation can be done by using various feedback control techniques. In this paper, the "state feedback" technique [9] is chosen for simplicity.

$$\dot{d}_j = u, \qquad u = -\lambda \hat{d}_j \tag{9}$$

where λ is the feedback gain.

Since the offset value is changed by the estimated offset values in the compensation system, the output equation Eq. (7), which is used in the fault-identification system, is no longer valid. And because there is no way to directly "measure" the compensated system in real time, the output values of output equations for the compensated system are obtained by the following two steps. (1) The compensated offset values are processed to obtain compensated sensor outputs, as the $m_{\mathrm{comp}\,j}$ shown in Eq. (10). And, the $m_{\mathrm{comp}\,j}$ would replace $m_{\mathrm{dc}\,j}$ in Eq. (7) to obtain new Z_p equations. (2) The new Z_{aux} equations remain the same as the ones shown in Eq. (7) although they maybe incorrect in the compensation system.

$$\begin{aligned}
m_{\mathrm{comp}\,j}(t) &= m_j(t) + d_j(t) = m_j(t) + d_j(o) - \lambda \int_0^t \hat{d}_j(\tau)\,\mathrm{d}\tau\\
&= m_{\mathrm{dc}\,j}(t) - \lambda \int_0^t \hat{d}_j(\tau)\,\mathrm{d}\tau
\end{aligned} \tag{10}$$

where, $d_j(o)$ is the offset value at initial time.

Since Z_{aux} equations are incorrect in the compensation system, the compensation system is no longer restricted to the constraint of only one faulty sensor in a sensor array. In turn, the proposed compensation algorithm can process multiple erroneous sensors producing faulty outputs simultaneously.

3.3. Real-time sensor drifts compensation

When the sensor errors are drifting, the previous proposed dc offset identification method would fail because the associated governing equations Eq. (6) is incapable of describing time-varying signals. In that case, the estimation failure can be attributed to the system modelling error. Since "fading memory" techniques and/or adaptive Kalman filter techniques [13,14] were developed to ensure the state convergence in the presence of model error, these methods can be used to work with the previously proposed estimation methods and to correctly estimate/compensate sensor drifts. This is done by adding the following algorithms into the standard EKF algorithms as shown in Eq. (8).

$$\begin{aligned}
&P^{-}(k+1) = \lambda_{\mathrm{f}}(k)P(k)\\
&\vdots\\
&M = H(k+1)P(k)H^{\mathrm{T}}(k+1)\\
&N = E[(z(k+1) - h(\hat{d}(k)))(z(k+1) - h(\hat{d}(k)))^{\mathrm{T}}] - R\\
&\lambda_{\mathrm{f}}(k+1) = \max\{1, \mathrm{trace}[N]/\mathrm{trace}[M]\}\\
&\vdots
\end{aligned} \tag{11}$$

where, λ_{f} is the forgetting factor. Because these methods ensure the state convergence by eliminating the effect of older data from current state estimation when they are no longer valid, the estimation accuracy of drifts is less than that of dc offsets.

4. Stability analysis of the real-time fault compensation method

The proposed real-time compensation method is essentially a task of stabilizing a nonlinear system using estimated state values. Due to a slightly difference between output equations of the estimation system and of compensation system, the "separation theorem" [15] can not be applied. Therefore, in this case, the stability analysis is proceeded for the observability of the system, the stability of the real-time fault identification, and the stability of the real-time fault compensation.

4.1. Observability of the system

In a dynamic system, the rank of the observability matrix and its associated singular values are indicators of a feasible observer design. Since the dc offset system is static, the observability matrix is solely composed of the Jacobian matrix of output equations [11] and can be processed as follows:

$$\begin{aligned}
&\mathrm{rank}(\nabla[\, Z_p \quad Z_{\mathrm{aux}1}\,]^{\mathrm{T}})\\
&= \mathrm{rank}\begin{bmatrix} C_{\mathrm{vot}\,1} \\ C_{\mathrm{vot}\,2} \\ d_1 C_{\mathrm{vot}\,1} + [\, C_{\mathrm{vot}\,1} d \quad 0 \quad \cdots \quad 0\,] \\ \vdots \\ d_{n+2} C_{\mathrm{vot}\,n+2} + [\, 0 \quad \cdots \quad 0 \quad C_{\mathrm{vot}\,n+2} d\,] \end{bmatrix}\\
&= \mathrm{rank}\begin{bmatrix} \begin{bmatrix} C_{\mathrm{vot}\,1} \\ C_{\mathrm{vot}\,2} \end{bmatrix} \\ \begin{bmatrix} C_{\mathrm{vot}\,1} d & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & C_{\mathrm{vot}\,n+2} d \end{bmatrix} \end{bmatrix}\\
&= \mathrm{rank}\begin{bmatrix} \begin{bmatrix} C_{\mathrm{vot}\,1} \\ C_{\mathrm{vot}\,2} \end{bmatrix} \\ \begin{bmatrix} C_{\mathrm{vot}\,1} m & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & C_{\mathrm{vot}\,n+2} m \end{bmatrix} \end{bmatrix}
\end{aligned} \tag{12}$$

The rank of the above matrix can be determined by examining the matrix $[C_{\text{vot}\,1}\, C_{\text{vot}\,2}]^{\text{T}}$ in the upper half, and the diagonal matrix in the lower half. The diagonal matrix is exactly the outcome of the voting equations. Therefore, exactly one element along the diagonal terms is zero. In turn, the diagonal matrix provides $n+1$ independent row vectors for the overall matrix. Furthermore, since $C_{\text{vot}\,1}$ and $C_{\text{vot}\,2}$ are two row vectors in a voting matrix, zero should not appear in the same column of these two vectors, and thus it provides one more independent row vector for the overall matrix. As a consequence, the rank of the observability matrix is $n+2$. Besides, the rank of the observability matrix does not depend on the system states, the system is "globally observable" [11].

4.2. Stability of the real-time fault identification

Although the EKF and the fading memory techniques are utilized to estimate state values, for simplicity of the stability check, the observer algorithm can be simplified as follows:

$$\dot{\hat{d}} = 0 + L\left(\begin{bmatrix} Z_p \\ Z_{\text{aux}} \end{bmatrix} - \begin{bmatrix} C_1\hat{d} \\ h_2(\hat{d}) \end{bmatrix}\right) = [\, L_1 \quad L_2\,] \begin{bmatrix} C_1(d-\hat{d}) \\ h_2(d) - h_2(\hat{d}) \end{bmatrix},$$

$$L = [\, L_1 \quad L_2\,], \qquad C_1 = [\, C_{\text{vot}\,1} \quad C_{\text{vot}\,2}\,]^{\text{T}} \tag{13}$$

where, L is the observer gain with proper dimensions and $h_2(\cdot)$ is the nonlinear function of Z_{aux} equations. The success of the real-time fault-identification is dictated by whether the estimation error, $e \triangleq d - \hat{d}$, would converge to zero. This can be shown by using the Lyapunov direct method [16].

$$\begin{aligned} &\dot{e} = -L_1C_1e - L_2[h_2(d) - h_2(\hat{d})] \\ &V = e^{\text{T}}Pe, \quad P:\ \text{positive definite} \\ &\dot{V} = e^{\text{T}}(-C_1^{\text{T}}L_1^{\text{T}}P - PL_1C_1)e - 2e^{\text{T}}PL_2(h_2(d) - h_2(\hat{d})) \end{aligned} \tag{14}$$

where, V is the Lyapunov function. The C_1 matrix has the dimension of $(n+2)\times 2$ and L_1C_1 has the dimension of $(n+2)\times(n+2)$. Therefore, there exists a proper L_1 that can arbitrarily assign two eigenvalues of the L_1C_1 matrix [12]. Consequently, all the eigenvalues of L_1C_1 can be smaller than or equal to zero. In that case, there exists a matrix Q, which is Hermitian and positive semi-definite (see Appendix A), such that:

$$\begin{aligned} &Q = -C_1^{\text{T}}L_1^{\text{T}}P - PL_1C_1 \\ &\text{Null}(Q) = \text{Null}(L_1C_1) = \text{Null}(C_1) = \text{Null}(C_{\text{voting}}) \end{aligned} \tag{15}$$

Therefore, Eq. (14) can be further processed as follows:

$$\begin{aligned} \dot{V} &= -e^{\text{T}}Qe - 2e^{\text{T}}PL_2\left(\begin{bmatrix} C_{\text{vot}\,1}d & 0 & 0 \\ & \ddots & \\ 0 & 0 & C_{\text{vot}\,n+2}d \end{bmatrix} d \right. \\ &\quad \left. - \begin{bmatrix} C_{\text{vot}\,1}\hat{d} & 0 & 0 \\ & \ddots & \\ 0 & 0 & C_{\text{vot}\,n+2}\hat{d} \end{bmatrix} \hat{d}\right) \\ &= -e^{\text{T}}Qe - 2e^{\text{T}}PL_2\left(\begin{bmatrix} C_{\text{vot}\,1}e & 0 & 0 \\ & \ddots & \\ 0 & 0 & C_{\text{vot}\,n+2}e \end{bmatrix} d \right. \\ &\quad \left. + \begin{bmatrix} C_{\text{vot}\,1}\hat{d} & 0 & 0 \\ & \ddots & \\ 0 & 0 & C_{\text{vot}\,n+2}\hat{d} \end{bmatrix} e\right) \end{aligned} \tag{16}$$

The first term in the right hand side of Eq. (16) is the quadratic term of the estimation error, while the second term is on the order of three. Therefore, there exists a region where the state values are close to zeros and thus the quadratic term in Eq. (16) dominates. As a consequence, the magnitude of the estimation error is decreasing everywhere except in the null space of Q.

Let e_{nu} be the estimation error in the null space of Q, the derivative of the Lyapunov function along the e_{nu} direction can be processed as follows:

$$\dot{V}|_{e=e_s} = -2e_{\text{nu}}^{\text{T}}PL_2 \begin{bmatrix} C_{\text{vot}\,1}\hat{d} & 0 & 0 \\ & \ddots & \\ 0 & 0 & C_{\text{vot}\,n+2}\hat{d} \end{bmatrix} e_{\text{nu}} \tag{17}$$

Since P is positive definite and the values of $\hat{d}$ are known, there exists a proper L_2 such that the derivative of Lyapunov function is negative when the estimation error is in the null space of Q. Combining conclusions from the last paragraph, the Lyapunov function in Eq. (14) is decreasing when the state values are close to zeros, and thus the real-time fault identification system is locally asymptotically stable.

4.3. Stability of the real-time fault compensation

As discussed before, in the compensation system, the offset values are changing and thus the values of Z_{aux} are unknown and replaced by zeros. The overall compensation system can be written as follows:

$$\begin{aligned} &\dot{d} = Bu \\ &y = [\, Z_p \quad Z_{\text{aux}}\,]^{\text{T}} = [\, (C_1d)^{\text{T}} \quad 0 \quad \cdots \quad 0\,]^{\text{T}} \\ &\dot{\hat{d}} = Bu + L\left(y - \begin{bmatrix} C_1\hat{d} \\ h_2(\hat{d}) \end{bmatrix}\right) = Bu + [\, L_1 \quad L_2\,] \begin{bmatrix} C_1(d-\hat{d}) \\ 0 - h_2(\hat{d}) \end{bmatrix} \\ &u = -K\hat{d} \end{aligned} \tag{18}$$

where, B is chosen to be the identity matrix with proper dimensions and the matrix K is for the feedback gain. The above system can be processed into a vector form:

$$\begin{aligned} &\begin{bmatrix} \dot{d} \\ \dot{\hat{d}} \end{bmatrix} = A_c \begin{bmatrix} d \\ \hat{d} \end{bmatrix} + \begin{bmatrix} 0 \\ -L_2h_2(\hat{d}) \end{bmatrix} \\ &A_c = \begin{bmatrix} 0 & -BK \\ L_1C_1 & -BK - L_1C_1 \end{bmatrix} \end{aligned} \tag{19}$$

The eigenvalues of A_c are determined by the eigenvalues of $-BK$ and $-L_1C_1$. Therefore, by choosing proper K and L_1, all the eigenvalues of A_c can be smaller than or equal to zero. Again, using a Lyapunov function, the stability of the compensated system can be shown.

$$\begin{aligned} &V = [\, d^{\text{T}} \quad \hat{d}^{\text{T}}\,]P \begin{bmatrix} d \\ \hat{d} \end{bmatrix}, \quad P:\ \text{positive definite} \\ &\dot{V} = -[\, d^{\text{T}} \quad \hat{d}^{\text{T}}\,]Q \begin{bmatrix} d \\ \hat{d} \end{bmatrix} + 2[\, 0 \quad -h_2^{\text{T}}(\hat{d})L_2^{\text{T}}\,]P \begin{bmatrix} d \\ \hat{d} \end{bmatrix} \\ &-Q = A_c^{\text{T}}P + PA_c \end{aligned} \tag{20}$$

According to Appendix A, the Q matrix is positive semi-definite and its null space is the same as that of the A_c. The null space of A_c can be found by the following:

$$\begin{aligned} &\begin{bmatrix} 0 & -BK \\ L1C_1 & -BK - L_1C_1 \end{bmatrix} \begin{bmatrix} d_{\text{nu}} \\ \hat{d}_{\text{nu}} \end{bmatrix} = 0 \\ &\Rightarrow \begin{cases} -BK\hat{d}_{\text{nu}} = 0 \\ L_1C_1(d_{\text{nu}} - \hat{d}_{\text{nu}}) = 0 \end{cases} \end{aligned} \tag{21}$$

Since BK is full rank, this leads to the following:

$$\begin{aligned} &\hat{d}_{\text{nu}} = 0, \qquad C_1d_{\text{nu}} = 0 \\ &\Rightarrow \dot{V}|_{d=d_{\text{nu}},\hat{d}=\hat{d}_{\text{nu}}} = 0 \end{aligned} \tag{22}$$

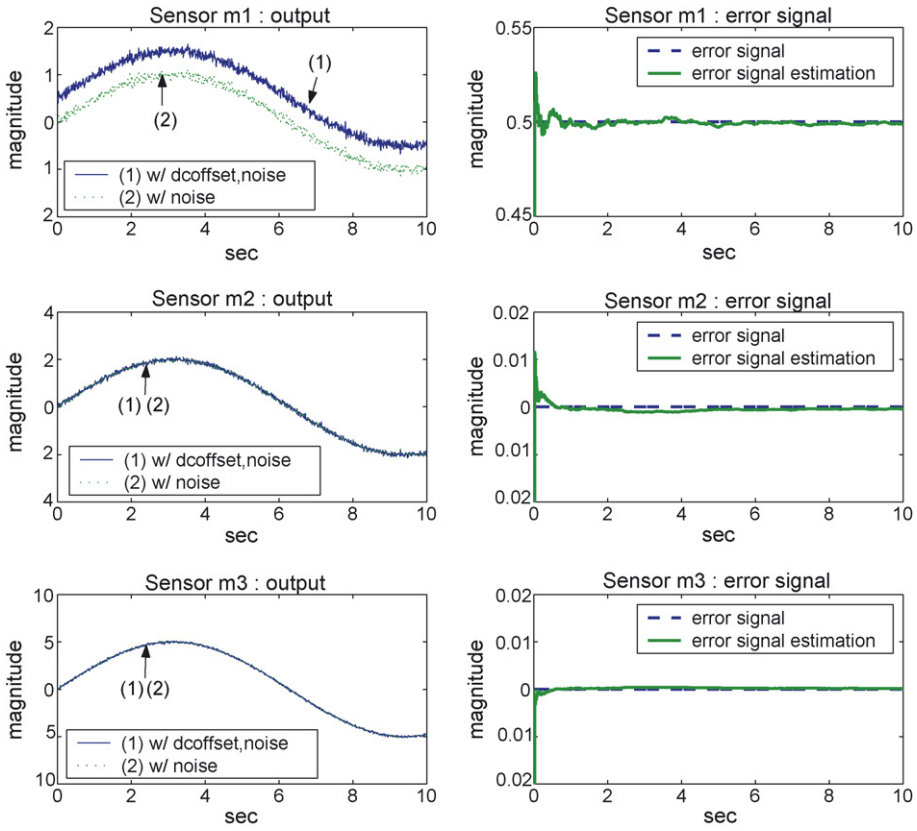

Fig. 1. Sensor outputs along a trajectory. The output of sensor m_1 is dc biased. When processed with the real-time fault identification algorithm, the values of the dc offsets can be correctly estimated. The estimation accuracy is less than 3.6×10^{-3}.

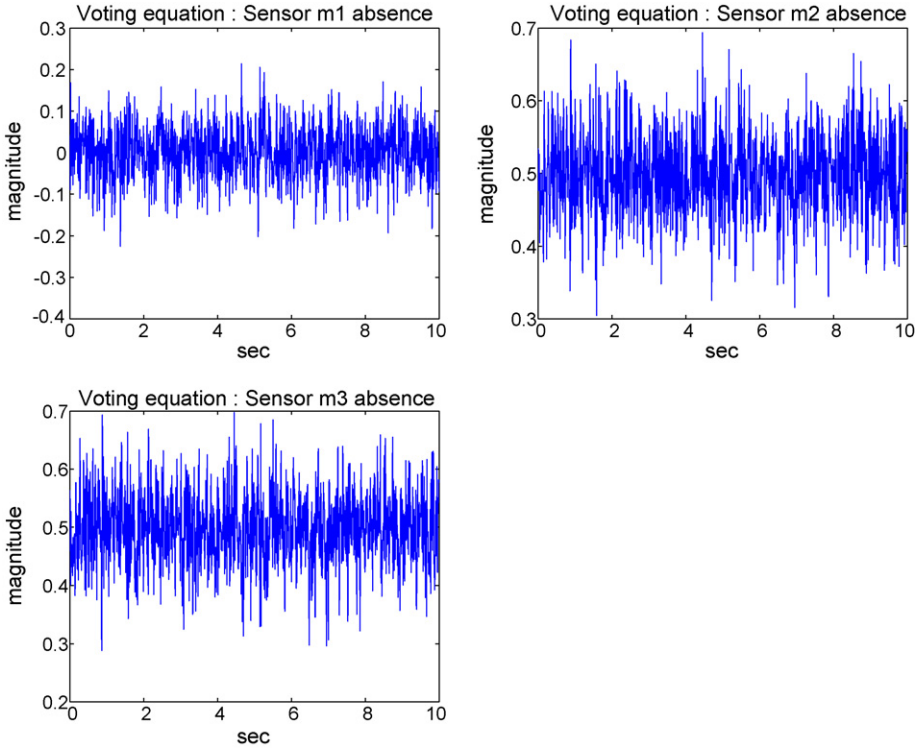

Fig. 2. The outputs of the voting equations along a trajectory. The first plot is zero mean, but the rest are all non-zero means.

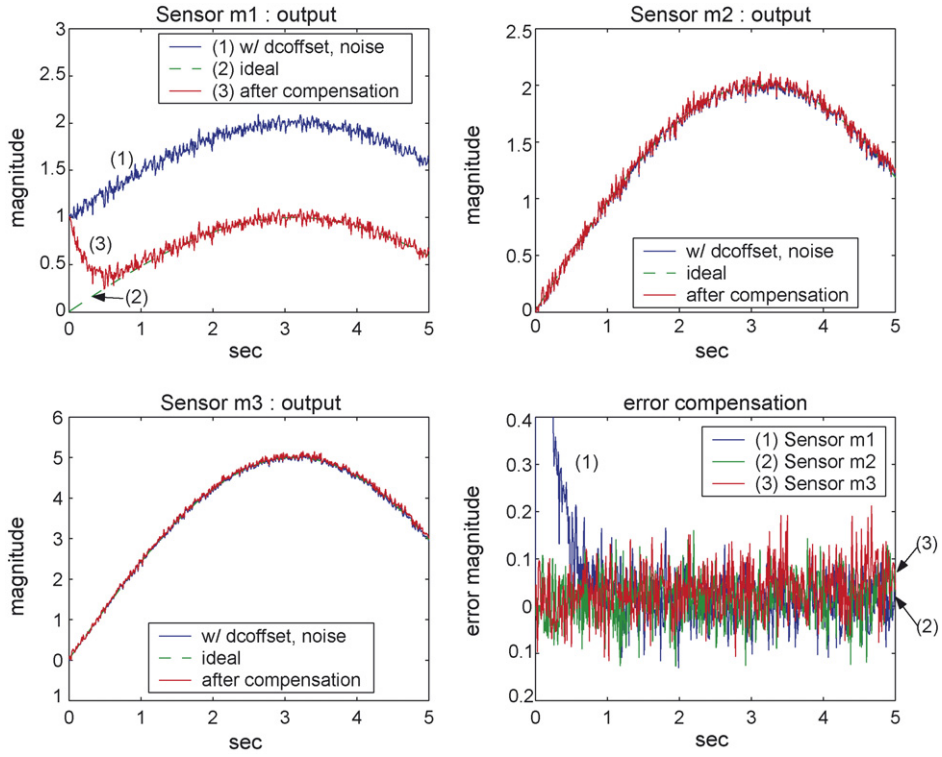

Fig. 3. Compensated sensor outputs converge to their respective correct values. Three compensated sensor outputs are biased at [0.004, 0.008, 0.02] with the standard deviations of [0.09, 0.08, 0.09].

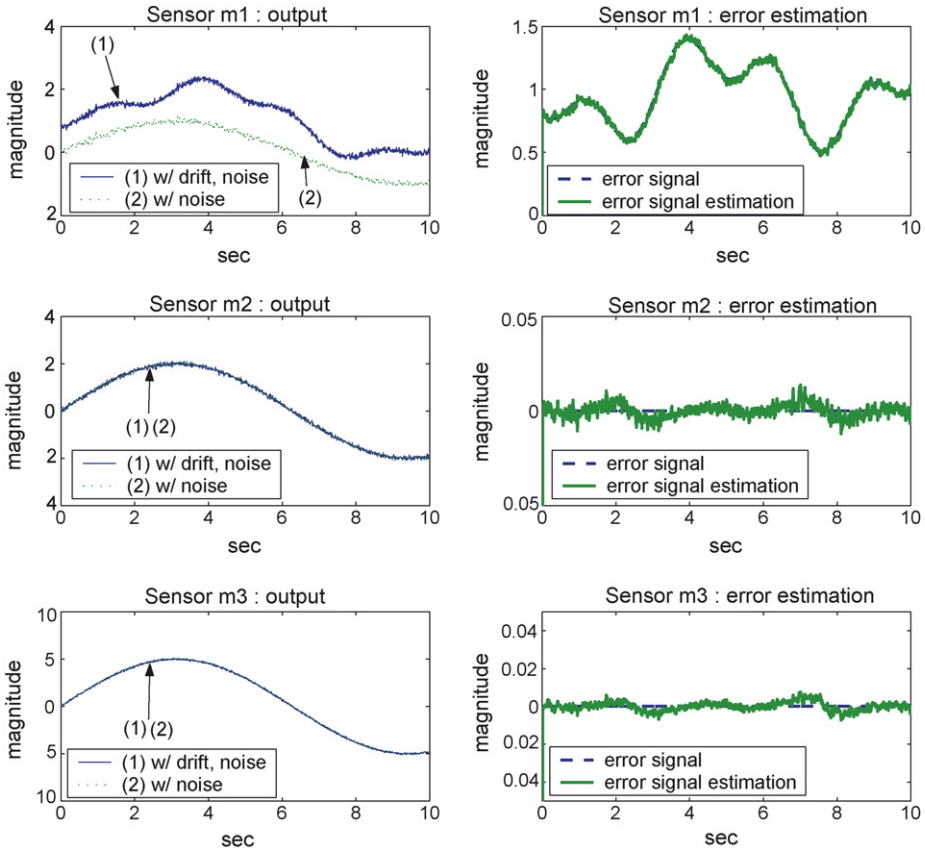

Fig. 4. Sensor outputs along a trajectory. The output of sensor m_1 is dc biased and drifting. When processed with the real-time fault identification algorithm, those drifts can be correctly estimated. The estimation accuracy is smaller than 0.026.

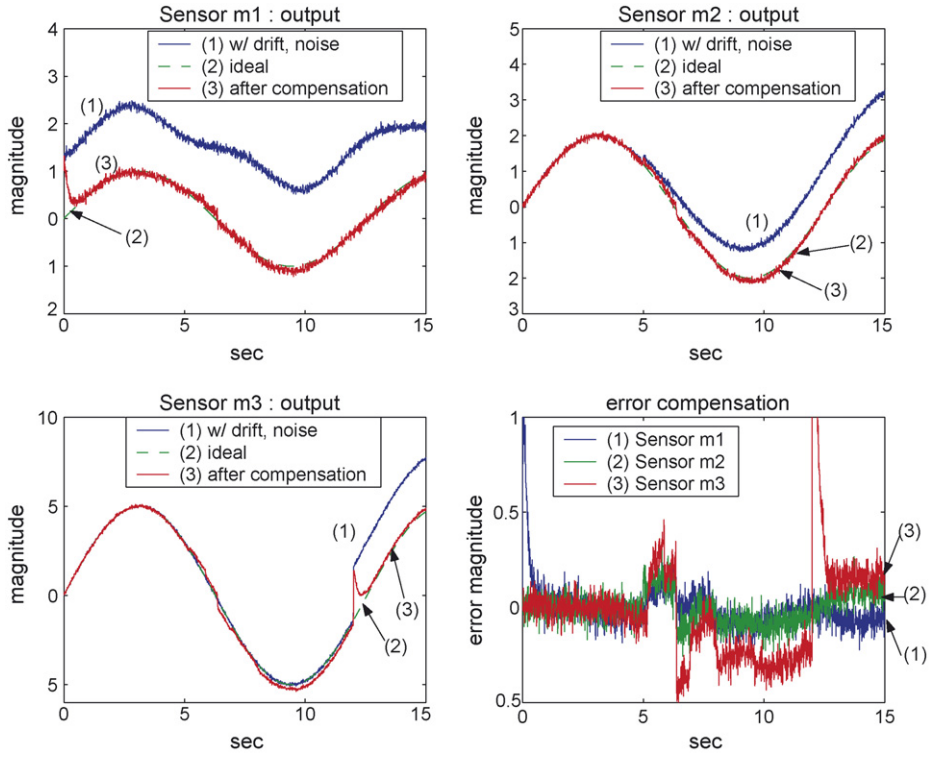

Fig. 5. Three sensor outputs are all erroneous. The m_1 and m_3 sensors are dc biased and drifting while the m_2 sensor is drifting. Compensated sensor outputs converge to their respective correct values.

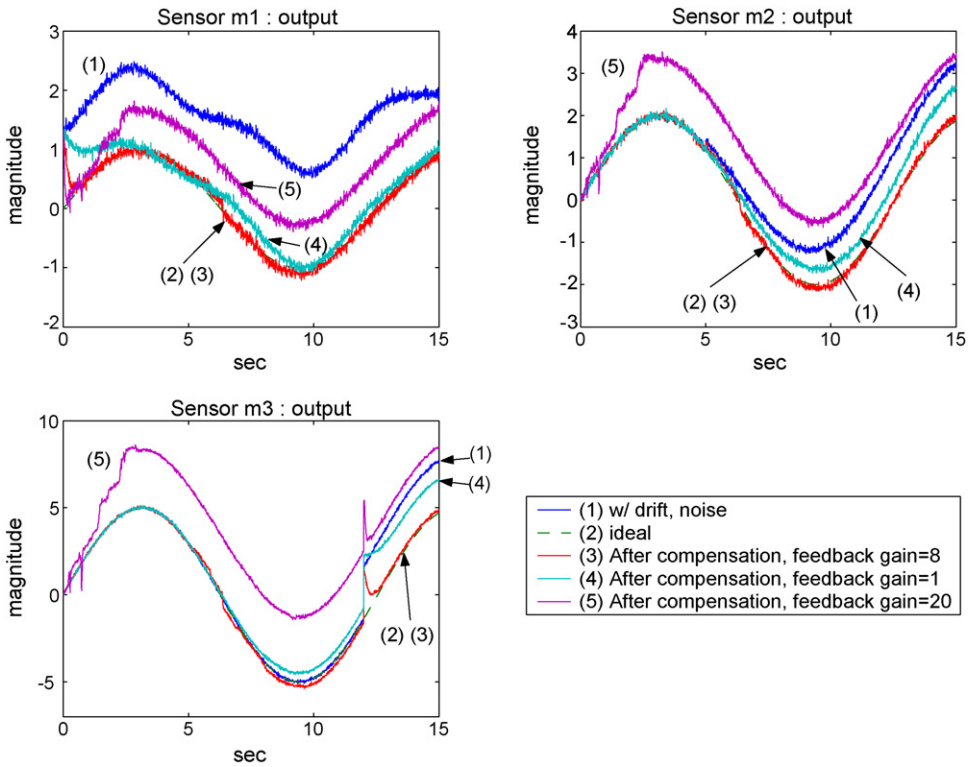

Fig. 6. The convergence of the sensor outputs varies depending on the feedback gain used in the compensation algorithm. The cases with the large feedback gain and small feedback gain both fail to decrease the sensor errors.

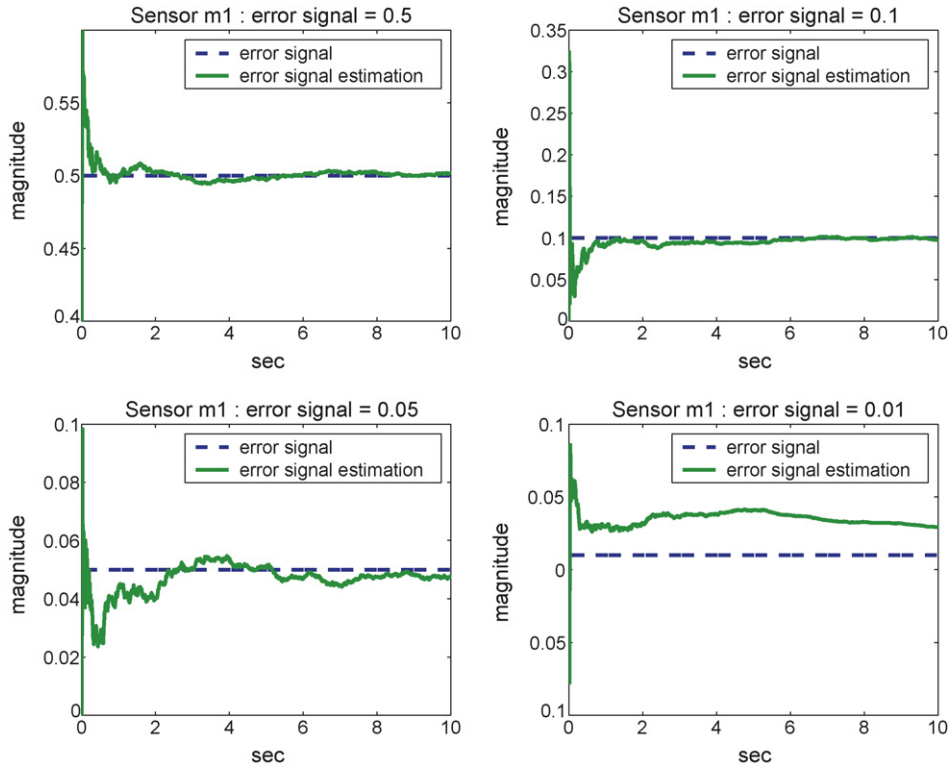

Fig. 7. The fault-identification method can estimate different magnitudes of the dc offsets. The minimum detectable dc offset is 0.1.

Therefore, when states are close to the origin, the derivative of the associated Lyapunov function is negative semi-definite. This implies that the error signal of each sensor can be reduced but may not decrease to zero. The sensor error would stop decreasing when it is in the null space of the C_1 matrix, which is the same as the null space of the C_{voting} matrix (Eq. (15)).

5. Simulation results

This section presents simulation results for an example design: the sensor m_1 is utilized to measure the behavior of the state x. By doing the sensor-redundancy design, the m_2 and m_3 sensors are deployed to measure the same state and to obtain two parity equations for the fault identification.

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} x + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}$$

$$\text{Parity equations}: \begin{cases} m_1 - 0.5m_2 \\ m_1 - 0.2m_3 \end{cases} \qquad (23)$$

$$\text{Voting equations}: \begin{cases} m_2 - 0.4m_3 \\ m_1 - 0.2m_3 \\ m_1 - 0.5m_2 \end{cases}$$

where, n_{1-3} is the sensor noise associated with each sensor. The state behavior to be measured is $\sin(0.5t)$. To demonstrate the capability of proposed compensation algorithm, one of the sensors, m_1 in this case, is arbitrary chosen to be dc biased at 0.5. Without loss of generality, the noise associated with each sensor measurement is assumed to be white with the same standard deviation of 0.1; this leads to a signal-to-noise ratio of five for the error signal estimation.

Three sensor measurements along a trajectory are shown in the left column of Fig. 1; all sensor measurements are contaminated by noise, and the outputs of the m_1 sensor is dc biased. The estimation of error signals is shown in the right column. According to the plots, the proposed estimation algorithm can correctly estimate the bias value for each sensor in real time. Thanks to EKF, the standard deviations of estimated bias values decrease to 3.6×10^{-3} for the last 5 s, which is only 1/30 of the sensor noise.

Fig. 2 shows the noise contaminated outputs of the voting equations of the sensor array. Only the first plot has zero mean and the remainder have non-zero means. The first plot is the case where the outputs of m_1 sensor is absent from the respective voting equation. At any time instant, none of the outputs of the voting equations is zero and thus the conventional fault identification method can not be used to identify the faulty sensor in real time. However, for the non-real-time approach, one can observe the outcomes of the voting equations over a period of time and determine that the m_1 sensor is dc biased.

Fig. 3 shows the sensor outputs after having been processed with the compensation algorithm. The feedback gain in the compensation algorithm is five. Although three sensor outputs seem to converge to their respective correct values after 0.6 s, a careful examination shows that three compensated sensor outputs are biased at the values of [0.004, 0.008, 0.02] with the standard deviations of [0.09, 0.08, 0.09]. Those bias values are 1/5 of the standard deviation of the sensor noise, and those standard deviations are the same as that of the sensor noise. Therefore, the proposed compensation algorithm reduce the bias values of incorporated sensors without deteriorating their sensing accuracy. Furthermore, those bias values converges to the null space of the voting matrix, which is $[1, 2, 5]^{\text{T}}$ in this example.

Fig. 4 shows sensor outputs for the case where one of the sensors experiences sensor drift. The error signals existed in the m_1 sensor is $1.5 - 0.3\sin(t) - 0.2\cos(2t)$. In this case, the change rate of the drift is four times faster than that of the correct signal to

be measured. The estimation of error signal is shown in the right column. As shown in plot, the proposed algorithm can correctly estimate the drift for each sensor in real time. This estimation is not biased with the standard deviations of [0.026, 0.003, 0.003] for three sensors. Although these standard deviations are 1/4 of the standard deviation of the sensor noise, they are about 10 times larger than that in the dc offset estimation. These results can be understood by the fact that the fading memory technique is ineffect to discard old measurements and to track a time-varying signal.

Fig. 5 shows the sensor outputs for the case where multiple faulty sensors are drifting. As shown in the plot, three sensor outputs are all drifting and these drifts are initiated at different time instant. The drifting signal with the m_1 sensor is $1.5 - 0.3\sin(0.5t) - 0.2\cos(t)$ starting at the time zero, the drifting signal with the m_2 sensor is $2(1 - e^{-0.1t})$ starting at the 5th second, and the drifting signal with the m_3 sensor is $3(1 - e^{-t}) + 0.1\sin(0.25t)$ starting at the 12th second. The feedback gain in the compensation algorithm is 8. According to this simulation result, the proposed compensation algorithm can correct all these drifting sensors. Note that, even to the compensated sensor outputs, this is the case of multiple erroneous sensors producing faulty outputs simultaneously.

Fig. 6 shows the convergence of the sensor outputs for various feedback gains, ranging from 1 to 20, used in the compensation algorithm. The sensor drift in this simulation is the same as in the previous simulation. As shown in the plots, the cases with feedback gain of 1 and 20 both fail to compensate for the sensor drifts. These results will be discussed in details in the next section.

Fig. 7 shows the case with the different dc offset values to be estimated. As evident from the plots, the proposed identification method fails when the bias value is smaller than 0.1, which happens to be the standard deviation of the sensor noise.

6. Discussion

In the case of only one faulty sensor existed in a sensor array, the proposed real-time fault-identification method can accurately estimate the error signal while the proposed real-time fault compensation method can reduce the error signals but the error may not decrease to zero. Therefore, in this particular case, there may exist an off-line compensation method that can correct the error signal accurately. The details of this off-line compensation method are under investigation.

The stability analysis (Section 4.3) indicates that the compensation system is a "local" stable system, meaning that the proposed compensation algorithm can reduce multiple sensor errors only when these errors are small. Intuitively, a large feedback gain in this compensation algorithm is preferred for two reasons: (1) it can compensate for the fast changing drift signals; (2) it can quickly decrease the magnitudes of current errors so that the constraint of local stability can be satisfied for the incoming sensor errors. According to the simulation results shown in Fig. 6, the feedback gain of 1 is too small to decrease sensor errors quickly. Consequently, the compensation system violate the local stability constraint at the 5th second, when the error of m_2 is initiated. On the other hand, from the experience with the LQG control method [12], the large feedback gain may lead to the oscillation of the system due to the noisy signals at the early stage of the control. This oscillation may just exceed the stability region and violate the local stability constraint. This is the case of feedback gain 20 shown in Fig. 6. Therefore, the highest change rate of the drifts, that can be compensated by the proposed method, is determined by the standard deviation of the sensor noise. The quantitative relation between the sensor noise and the change rate of the drifting signal is yet to be determined.

When lowering the magnitude of the dc offset in the sensor outputs, the proposed fault-identification method fails to identify the offset value at around 0.1, which happens to be the standard deviation of the sensor noise. Ideally, for a linear system, the Kalman filter algorithm can correctly estimate states for the state values less than the standard deviation of the sensor noise. Therefore, it could be the linearization error, introduced by the EKF algorithm, that limits the detection accuracy of the proposed fault identification design. However, more research is necessary to investigate this.

7. Conclusion

This paper presents design procedures and stability analysis in details for a fault-tolerant sensor system. This design uses redundant sensor components together with state estimation techniques (EKF) to estimate the values of error signals. It then uses state feedback techniques to decrease the error signals in real time and to satisfy the local stability constraint. In such, the fault-tolerant system can perform robust measurements even when all of its incorporated sensors are noisy and drifting.

The analysis indicates that the proposed identification method can estimate a time-varying error signal accurately when there is only one faulty sensor in a sensor array. But, it would fail when multiple sensors are erroneous. One the other hand, when the local stability constraint is satisfied, the proposed real-time compensation method can decrease the error signals for multiple erroneous sensors but those errors may not converge to zero. The errors stop decreasing when they are in the null space of the associated voting matrix. Furthermore, the highest change rate of the drifts, that can be compensated by this method, is determined by the standard deviation of the sensor noise.

The simulation results of a design example indicate that the minimum detectable dc offset value is equal to the standard deviation of the sensor noise. Once the offsets are detected, the estimation accuracy becomes 3.6×10^{-3}. After processed with real-time compensation, three compensated sensor outputs are biased at [0.004, 0.008, 0.02] with the corresponding standard deviation of [0.09, 0.08, 0.09]. The proposed compensation method can compensate sensor drifts without deteriorating the sensing accuracy.

Appendix A. Lyapunov functions for ISL stable systems

This proof is listed here as a appendix because a similar theorem, shown in may textbooks [17], was for the asymptotically stable system but not for the stable in-the-sense-of-Lyapunov (ISL) system [9].

Theorem. *For a linear, ISL stable system:* $\dot{x} = Ax$, $\mathrm{eig}(A) \leq 0$. *Given a positive semi-definite matrix Q and Q has the same null space as A, there exists a unique positive definite matrix P such that* $A^{\mathrm{T}}P + PA = -Q$.

Proof.

$$\forall x_0 = x(t)|_{t=0} = x_{a0} + x_{\mathrm{nu}}$$
$$Ax_{\mathrm{nu}} = Qx_{\mathrm{nu}} = x_{\mathrm{nu}}^{\mathrm{T}}Q^{\mathrm{T}} = x_{\mathrm{nu}}^{\mathrm{T}}Q = 0$$
$$x(t) = e^{At}x_0 = e^{At}x_{a0} + x_{\mathrm{nu}}$$

where, x_{nu} is the linear combinations of the eigenvectors of the A matrix, with their corresponding eigenvalues being zeros.

$$x_0^T P x_0 - x_\infty^T P x_\infty = -\int_0^\infty d(x^T P x)$$

$$= -\int_0^\infty \frac{d(x^T P x)}{dt}\, dt (x_{a0} + x_{nu})^T P (x_{a0} + x_{nu})$$

$$-x_{nu}^T P x_{nu} = -\int_0^\infty x^T (A^T P + PA) x\, dt$$

$$x_{a0}^T P x_{a0} + 2x_{a0}^T P x_{nu} = \int_0^\infty x^T Q x\, dt = \int_0^\infty (e^{At} x_{a0} + x_{nu})^T$$

$$\times Q (e^{At} x_{a0} + x_{nu})\, dt = \int_0^\infty (e^{At} x_{a0})^T Q (e^{At} x_0)\, dt$$

Let $P = \int_0^\infty e^{A^T t} Q\, e^{At}\, dt$,

$$\Rightarrow 2x_{a0}^T P x_{nu} = 2\int_0^\infty (e^{At} x_{a0})^T Q\, e^{At} x_{nu}\, dt$$

$$= 2\int_0^\infty (e^{At} x_{a0})^T Q x_{nu}\, dt = 0$$

Therefore, $P = \int_0^\infty e^{A^T t} Q\, e^{At}\, dt$ is one of the solutions for the equation $A^T P + PA = -Q$. Furthermore, since Q is positive semi-definite, Q can be written as $Q = H^T H$.

$$x_{a0}^T P x_{a0} = \int_0^\infty (e^{At} x_{a0})^T Q (e^{At} x_{a0})\, dt = \int_0^\infty \|H\, e^{At} x_{a0}\|_2^2\, dt > 0$$

$\Rightarrow P$ is a positive definite matrix.

The above proof shows the positive definiteness of the P matrix. The "uniqueness" property can be proved with a little bit twist. Since it is not the main interest here, this part of the proof is ignored. □ □

References

[1] C.Y. Lee, G.B. Lee, MEMS-based humidity sensors with integrated temperature sensors for signal drift compensation, Sensors, Proceeding of IEEE (2003) 384–388.

[2] Y.T. Lee, H.D. Seo, Compensation method of offset and its temperature drift in silicon piezoresisive pressure sensor using double wheastone-bridge configuration, Transducers'95 (1995) 570–573.

[3] A.D. Pouliezous, G.S. Stavrakakis, Real Time Fault Monitoring of Industrial Processes, Kluwer Academic Publishers, 1994.

[4] H. Benitez-Perez, F. Garcia-Nocetti, H. Thompson, Fault classification SOM and PCA for inertial sensor drift, Sensors, Proceedings of IEEE (2005) 177–182.

[5] T.B. Tang, A.F. Murray, Adaptive, integrated sensor processing to compensate for drift and uncertainty: a stochastic "neural" approach, IEE Proceedings of Nanobiotechnology 151 (1) (2004) 28–34.

[6] U. Krogmann, Failure management in spatio-temporal redundant, integrated navigation and flight control reference systems, Position Location and Navigation Symposium (1990) 330–337.

[7] J. Gilmore, R. McKern, A redundant strapdown inertial reference unit (SIRC), Journal of Spacecraft and Rocket 9 (1) (1972) 39–47.

[8] A. Ray, R. Luck, An introduction to sensor signal validation in redundant measurement systems, Control Systems Magazine IEEE (1991) 44–49.

[9] C. Chen, Linear System Theory and Design, Library of Congress Cataloging in Publication Data, 1984.

[10] Y. Bar-Shalom, X.R. Li, T. Kirubarajan, Estimation with Applications to Tracking and Navigation, John Wiley & Sons, 2001, p. 371.

[11] R. Hermann, A. Krener, Nonlinear controllability and observability, IEEE Transactions on Automatic Control AC-22 (1977) 728–740.

[12] F.W. Fairman, Linear Control Theory: The State Space Approach, John Wiley & Sons, 1998, pp. 147–166.

[13] Q. Xia, M. Rao, Y. Ying and X. Shen, Adaptive Fading Kalman Filter with an Application, IFAC Automatica 30 (8) (1994) 1333-1338.

[14] G. Hu, W. Chen, Y. Chen, D. Liu, Adaptive kalman filtering for vehicle navigation, Journal of global position systems 2 (1) (2003) 42–47.

[15] M. Vidyasagar, On the stabilization of nonlinear systems using state detection, IEEE Transactions on Automatic Control AC-25 (3) (1980) 504–509.

[16] J.J. Slotine, W. Li, Applied Nonlinear Control, Prentice Hall, 1991, pp. 40–97.

[17] H.K. Khalil, Nonlinear Systems, third edition, Prentice Hall, 2000, pp. 111–181.

Biographies

Tsung-Lin Chen received his B.S. and M.S. degrees in power mechanical engineering from National Tsing Hua University, Shinchu, Taiwan in 1990 and 1992, respectively. He received his Ph.D. degree in mechanical engineering from the University of California, Berkeley in 2001. From 2001 to 2002, he joined Analog Devices Inc. as a MEMS design engineer. Since 2003, he has been with the Department of Mechanical Engineering, National Chiao Tung University, Shinchu, Taiwan. Currently, he is an assistant professor. His research interests include MEMS and controls.

Ren-Zhi You received his M.S. degree in mechanical engineering form National Chiao Tung University, Shinchu, Taiwan in 2006. Currently, he serves in Army in Taiwan.